MASTER IN COMPUTER SCIENCE

Argumentation for security
a comparative study

François, Stéfanie

Award date:
2010

Awarding institution:
University of Namur

FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX NAMUR
FACULTÉ D'INFORMATIQUE

Argumentation for Security
A Comparative Study

Stéfanie François

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU GRADE DE MASTER
EN SCIENCES INFORMATIQUES À FINALITÉ SPÉCIALISÉE

Année Académique 2009-2010

Abstract

Argumentation is primarily a field of study and a philosophical discipline which has been widely taught for centuries. This discipline allows for the development and improvement of students' critical thinking. It has also been studied in law schools as a part of training of future lawyers in defence of their client, especially in preparation and development of the most decisive arguments in favour of the latter. Many software packages have been developed to enable users to practice visual argumentation and to create his or her own argument maps.
In this thesis, we have attempted to reconcile these software tools, designed to create visual arguments, with the field of computer security. This thesis explores the possibility of creating an argument map of security requirements using visualization argument software tools. More specifically, this thesis presents a comprehensive assessment of five different tools; Araucaria, ArguMed, Argumentative, Athena and Rationale, all designed to create visual argument maps. The presented assessment was based on a design of the "security requirement" argument and explores, for each tool, five different perspectives: the user interface, available features, the documentation and error handling, the notations used to represent the argument and the argumentation ontology underlying each tool. Finally, a discussion interprets the results and the relevance of these tools in the field of security requirements.

Keywords : *Argumentation Theory, Security Requirements, CSAV, Argument Creation, Argument Visualization, Araucaria, Athena, ArguMed, Argumentative, Rationale*

Résumé

L'argumentation est avant tout un champ d'étude et une discipline philosophique largement enseignée depuis des siècles. Cette discipline permet le développement et l'amélioration du sens critique de l'étudiant. Elle est tout aussi étudiée dans les écoles de droit pour apprendre aux futurs avocats à défendre leur client en préparant et en avancant les arguments les plus décisifs en faveur de ce dernier. De nombreux logiciels ont été développés pour permettre à tous de pratiquer l'argumentation de manière visuelle, ils permettent à l'utilisateur de créer ses propres arguments sous la forme d'un diagramme.
Dans cette thèse, nous avons tenté de rapprocher ces outils logiciels du domaine de la sécurité informatique. Cette thèse explore donc la possibilité de créer la visualisation d'exigences de sécurité sous forme d'un diagramme à partir d'outils logiciels d'argumentation visuelle. Plus particulièrement, cette thèse présente une analyse complète de cinq différents outils; Araucaria, ArguMed, Argumentative, Athena et Rationale, tous conçus pour créer des arguments visuels sous la forme d'un diagramme. L'évaluation proposée est basée sur la conception d'un argument de type "exigence de sécurité" et aborde chacun de ces outils sous cinq angles différents : l'interface utilisateur, les fonctionnalités disponibles, la documentation et la gestion des erreurs, les notations employées pour représenter l'argument et le style d'argumentation sous-jacent à chaque outil. Finalement, une discussion revient sur les résultats obtenus et la pertinence de ces outils dans le domaine de conception des exigences de sécurité.

Mots-clé : *Théorie de l'argumentation, Exigences de sécurité, CSAV, Construction d'arguments, Visualisation d'arguments, Araucaria, Athena, ArguMed, Argumentative, Rationale*

Acknowledgements

I would like to acknowledge all the following people who contributed to making my thesis process a valuable experience.

Prof. Pierre-Yves Schobbens, my supervisor at the University of Namur, for helping me with fixing of the objectives of my research, for providing me with the required documentation, for guiding me in writing of this thesis and for correcting its different versions.

Prof. Dr. Charles B. Haley and Dr. Yijun Yu, for their hospitality during my internship in the Jennie Lee building at the Open University in Milton Keynes, for discussions about the argumentation topic, for bringing the subject of this thesis to my attention and for their advices about the research process.

Furthermore I would like to acknowledge the following people for their help with my everyday life during my internship: Pooja, Zdenek, Sunita, Jenifer, Laura, Alexandra, Eva, Minh, Rose, Lucas and particularly Lionel and Daniele who were always there for me at any time.

I would like to give credits to the academic and non academic staff members at the University of Namur for sharing parts of their knowledge and for allowing me to become who I am today.

Special thanks goes to my friends for all the wonderful moments we have shared during our student years, but also for their support in hard times.

Last, but not least, I would like to thank my father, my sister Céline, and my fiancé Maxime for providing me with great support, encouragement and understanding throughout this entire process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The argumentation is primarily a field of study and a philosophical discipline widely taught for centuries. This discipline allows the development and improvement of students' critical thinking. It is also studied in law schools to teach future lawyers to defend their client in preparing and by advancing the most decisive arguments in favor of the latter. Many software packages have been developed to enable users to practice visual argumentation, i.e. with these software tools, user can create his or her own argument maps.

The subject of this thesis is to convey to the reader the results of the assessment made on the basis of five visualization argumentation software tools and a specific task : achieving an argumentation map based on security requirements. The assessment criteria focus on the software user interface, its convenient functionality, the level of available documentation for the user, the notations and the underlying ontology of the argumentation model.

Investigation of the evolving security requirements implemented using these visualization argumentation software tools was my main interest at the beginning of writing of this thesis, but the lack of major differences between this type of argumentation and a traditional argumentation caused a limitation into the central position played by the design of security requirements. The security requirements were represented in the form of declarative sentences as most traditional components of arguments are, with argument and counterarguments supporting or rejecting the main claim. The content of argument was different, but the form was almost the same. Major interest was naturally inclined towards the argumentation ontology underlying each tool, bearing less interest in the type of the security requirements expressed by the content of the argument.

As a compromise in this work we have chosen to assess each tool on a basis of five relevant criteria, but without concentrating too much on the specific nature of the argument. Despite all this, the connection between visualization argumentation tools and the design of security requirements has been kept and, therefore, provides certain guidelines for this work. We discuss about this problem with the aim to show what is the impact of the conception of argumentation related to security requirements.

During my internship at the Open University, I worked on the development of a visualization model of the security requirement argument. I have achieved very concrete results, I defined the main concepts I needed to create the visualization, I described the syntax of the language and the meta-model with relationships between the concepts underlying the model of my argumentation visu-

alization. However, after my return in Belgium, I realized the mistake I was making when I wanted to define a visualization of an argument, but without exploring the existing tools on the market. That is why the work I had done during the internship is not related to the work of this thesis. The initial goal of this work was to carry out the assessment described above in order to improve my implementation of the visualization, but then my interest in the argument mapping tools has risen and consequently I deepened my assessment and left out the implementation of the visualization for maybe a future work. This thesis is an assessment of some software tools, but more specifically, it is the examination of those existing tools that I did not take the time to explore during my internship. I hope this assessment will help other people in the field.

1.1 Structure

My knowledge and my understanding of the subject being what it is after the last few months that I had worked on it, I am glad to present you the outline of my thesis which is divided in four parts as follows.

Part I introduces the background of the work by providing the foundations of the argumentation theory in Chapter 2. Chapter 3 falls entirely within the subject of the Computer-Supported Argument Visualization by explaining as the central point benefits that a visual representation of an argument implies as the focal point. This first part ends with a final chapter totally different from the first three chapters dealing with the argument. Indeed, in Chapter 4, we give a brief overview of the foundations of security requirements so that in the next part we will able to build a bridge between these two completely different subjects.

Part II presents the contribution of this work. Chapter 5 contains a complete overview of the assessment process and gives all necessary information about assumptions and different criteria used during the assessment. Chapter 6 presents the results of the assessment for each argumentation tool.

Part III is devoted to discussion about the results obtained, providing answers, but also raising new questions. Chapter 7 gives some recommendations regarding the different criteria we studied. Discussion then focuses on the place of security requirements and conccludes with a review of the assessment process.

Part IV finally presents the conclusions of this work and a potential future work in the field.

Part I

Background

Chapter 2

The Principles of Argumentation

> Mother to daughter : *Forget about vacationing all by yourself. No way ! When your sister was fourteen, we didn't let her go either.*[14]
>
> Michael Schumacher is the cheekiest F1 driver in the history of the sport because, because he scored more points than any other driver in 7 seasons!
>
> *Since 1976, states [in the United States] have executed 612 people, and released 81 from death row who were found to be innocent. Is there any reason to believe that the criminal justice system is more accurate in non-capital cases? If the criminal justice system makes half the mistakes in non-capital cases that it makes in capital cases, thousands of innocent people live in our prisons.*[33]

Here are some examples of arguments. Who is not familiar with argumentation ? Argumentation is used by everyone in different context as in newspapers or in discussions of everyday life (see for example [14]). Reasoning by argument and counterargument is present in verbal and written communication forms. We react all the time to arguments put forward by others and we advance arguments intended to defend, debate or explain our thoughts, feelings or actions.

In this chapter, we outlined foundations of the argumentation theory. The first section presents our investigations of the benefits to study the argumentation. Section 2.2 defines more precisely the concept of argumentation theory and its three different perspective that we discuss in the three following sections which are the analytic, the rhetoric and the dialectic. Section 2.3 about the analytic side of the argumentation results from the logic basis of the argumentation. The rhetoric part presented in Section 2.4, introduces the work of Aristotle, the different types of oratory, the improvement of Perelman about the audience and the Chart Method designed by John Wigmore. Finally, the dialectic principle is discussed in Section 2.5 with a large description of the Toulmin's Model of Argumentation which will take us to the next chapter on the arguments presented under a visual map form.

2.1 The Study of Argumentation

The study of argumentation is at the very core of human learning. Already in the 5th century AD, the argumentation theory was included in the classical education curriculum. The aim is to develop the

student's rational thought and intellectual capabilities and so to prepare him for the pursuit of science in the strict sense of the term, i.e. the combination of philosophy and theology known as scholasticism [61].

One century ago, the Logical Positivism movement of scientists was convinced that the world and the universe were deterministic, governed by laws of cause and effect that science could understand well enough to be able to predict and control the world with a deductive reasoning from the world observation. After the Second World War, this movement collapsed. Another way of thinking is then born, more hesitant. Nowadays, scientists are faced with new hesitations between theory and observation. They have to support their conviction with arguments to persuade others to develop new theories.

In recent years, the understanding of the theoretical properties of different argumentation logics increased considerably in many different fields [11]. The argumentation theory is an excellent example of interdisciplinary area of research. It extends to several fields including philosophy, communication studies, linguistics, law and psychology. With the emergence of Computer Science, contributions and ideas originating from many of these disciplines have influenced the interest of a computational theory of argumentation in artificial intelligence.

The study of argumentation focuses on how people argue, or should argue, their opinion in controversial contexts. The attention of argumentation holds on questions about concrete methods to build a correct argument [7]. These questions are straddling on different themes about argumentation process as : defining the main components of an argument, highlighting interaction between these components, distinguishing invalid arguments, identifying rules of argumentation processes, determining conditions under which further discussion is redundant, analysing the typical structures of argument components etc.

Argumentation can be also considered as a set of propositions relying on evidences. The evidence of propositions used to debate is an other interest in the study of argumentation. Evidence for a particular proposition could be defined as a special piece of information that, by its support to the proposition, increases the estimate of the probability that the proposition is true [19]. Evidence could be gathered at many sources of information, depending the proposition to support. Evidence is also defined more specifically in the legal field as something legally submitted to a tribunal to ascertain the truth of a matter [13].

Finally, a last interest of this discipline is to understand the processes by which participants engaging in debate may advance their respective positions and predetermine contrary stances and arguments. Most discussions are specific to the context, namely to the moment when they are shown, to the audience that listens and makes his own judgements, to the expression of arguers etc.

To conclude this overview on the investigations of the study of argumentation, it is worth noting that, as in any discipline, the best way to completely understand and be able to teach the discipline is to study deeply the principles of its operation and its processes.

2.2 The Argumentation Theory

The Fig 2.1 shows the three different perspectives of argumentation theory followed in this chapter to give to the reader a better understanding of the argumentation process, and its brief historical view.

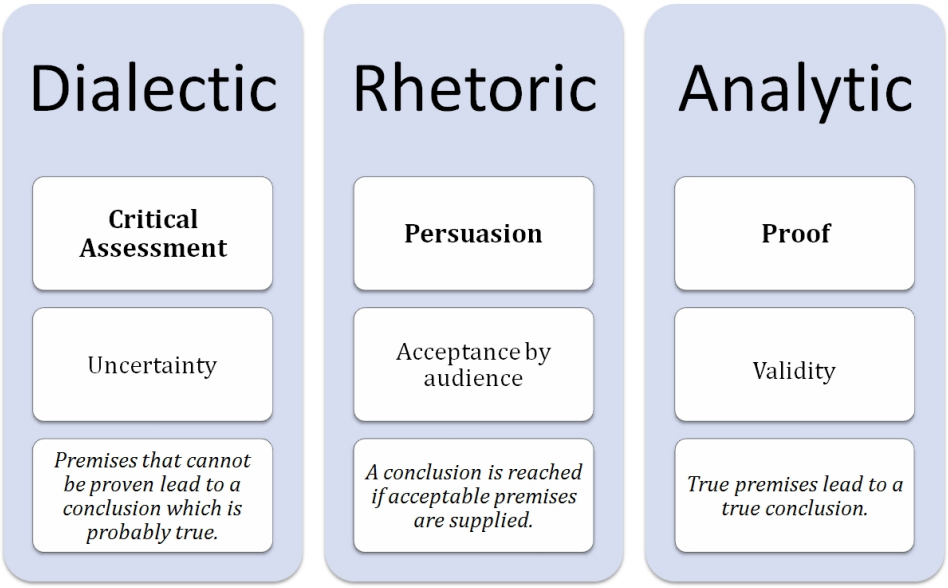


Figure 2.1: Fields of argumentation

[14] gives the definition of the argumentation as :

> *"Argumentation is a verbal and social activity of reason aimed of increasing (or decreasing) the acceptability of a controversial standpoint for the listener or reader, by putting forward a constellation of propositions intended to justify (or refute) the standpoint before a rational judge."*

In other words, in an argumentation between two or more interlocutors, each advances arguments to defend their own standpoint about the subject of the debate, then they start weighing up the different considerations, contemplating the pros and cons of their own ideas and other's. The ultimate objective is to increase the acceptability of a controversial standpoint until convincing the audience in justifying the *pro-arguments* and refuting the *contra-arguments*.

Argumentation can be seen as the studied method to correctly and completely answer the question of the arguer : *Why this standpoint is the good one, why this standpoint is valid ?*

A first approach of the argumentation is the analytic perspective close to the deductive reasoning in logic. The goal is to ensure the validity of the reasoning. If the reasoning is certified to be valid, then the conclusion will be accepted. The second approach takes into account the role of the rhetoric in argumentation; rhetoric focuses on the way used to persuade an audience. Finally, a third approach focuses on the dialectic perspective. The dialectic aims attention at the truth of a theory or opinion by

a critical assessment. These last two perspectives are described as *the warp and woof of argumentation analysis* [52].

Let us look closer to these three perspectives which describe the operation of argumentation.

2.3 The Analytic Argument or Mathematical Proof in Argumentation

The objective of an analytic argument focuses on the validity of the evidence, on the proof. Evidence is a true statement used with inference rules. A simple sentence which just summarizes this idea is : *"True premises lead to a true conclusion"*.
Analytic arguments are described as :

Rational :	Consistent with or based on reason; logical.
Observable :	A measurable property of a physical system.
Measurable :	Able to be measured; perceptible or significant.
Timeless :	Independent of time; eternal.

Analytic argument implies the realisation of discussion in a closed world.

2.3.1 The Analytic Logic of the Syllogism

Syllogisms are arguments that take several parts, typically with two statements which are assumed to be true (or premises) that lead to a conclusion.

The canonical syllogism first described by Aristotle, with the third statement derived from the previous two, is:

Humans are mortal.
Greeks are human.
Therefore, Greeks are mortal

Syllogisms are particularly interesting in persuasion as they include assumptions that many people accept which allow false statements or (often unspoken) conclusions to appear to be true. The most important idea to understand in syllogism is the difference between validity and truth. Reaching the conclusion that a syllogism is valid, implies that its form is valid, not necessarily its content; the truth of the statement's content does not matter.

The general form of a syllogism in theory is given by three statements :

Major premise	A general statement.
Minor premise	A specific statement.
Conclusion	Based on the two premises.

For an arguer, knowing all the possible valid forms of a syllogism and being able to find out the wrong forms can help him to counteract opponents.

Fallacies - Realisation that the way humans argue is not like formal logic.

2.3.2 Relation With Mathematical Proof

Argumentation is not simply a logical reasoning formulated in natural language. It is essential to understand similarities and differences between the argumentation theory and the traditional concepts of logical reasoning and mathematical proof in order to better position the argument in relation to mathematical reasoning.

The logical reasoning is the basis of the development of argumentation models in many disciplines [7]. Some structural elements of mathematical reasoning have proven to be useful in argumentation. For example, formulae defined for expressing assertions in mathematical proof are often used in argumentation, and it is also the case for axioms, the accepted bases on which to build theorems. Other elements such as mathematical templates or procedures also assist to derive new theorems from existing ones. These elements imported from logic and formal deductive reasoning brought modelling and analysis argumentation along a powerful basis. Furthermore, those are used in argumentation models for their ease of use.

However, argumentation in political debates, controversy of ethical principles or deliberation in judicial settings, is radically different in nature to the formal methods of mathematical reasoning. Two significant differences totally distinguish the mathematical proof from the argument.

The most significant difference concerns the nature of the arguments. Argument is defeasible. The reasoning based on an incomplete and uncertain context is accepted just for a while by the audience. Subsequently, after changes in view points or the awareness of information not previously available, the argument fails to convince. Conclusions of arguments are never definite. The argument never goes into a state where the defeasibility property is no longer applicable; it may always be challenged at some point. In mathematical reasoning, at the opposite, premises are defined in terms of ground propositions with a closed-world assumption. Conclusions are final and do not change with new information. A conclusion proved once, is accepted forever.

The second difference between a formal proof and a persuasive argument is related to their objective. A mathematical proof aims at demonstrating logically a conclusion. Much emphasis is placed on the correctness of the proof. The audience judges the mathematical reasoning as correct or not. If the reasoning is not correct, the conclusion has no value, otherwise it is accepted by all. In argumentation, the persuasive argument is used to convince of a well-founded conclusion. The argument takes into account the subjectivity of the audience. In a debate, the audience has a different acceptance rate, depending on the views, attitudes or prejudices of each.

2.4 The Rhetoric Argument or the Art of Persuasion

> "Classical rhetoric can be defined as the science of persuasion through the use of language, the discipline which studies the way a speech can be made more effective through the use of devices which can be identified, classified and learnt."[**?**]

Rhetoric is therefore a practical discipline, a "technique", as it is called by Aristotle. *"Ars bene dicendi"*, or the art of good talking is the shorter classical definition of rhetoric, insists on this practical side of rhetoric. Equally important is the notion that rhetoric is concerned above all with oral discourse–not with written language, which is the main concern of poetics. Also, there is no question

of inspiration in rhetoric. Being a practical technique, it is concerned with controllable and predictable means of construction and persuasion, not with speculations about the origin of beauty or invention. The emphasis is all the time on practice, pre-established exercise, and constant work.

Initially, we discuss in this section rhetoric according to Aristotle and then we describe the contribution of the New Rhetoric of Perelman.

2.4.1 Aristotle

Like most works of the Greek philosopher Aristotle, Aristotle's Rhetoric was redacted from a collection of his teachings at the Lyceum, the school he founded in Athens in 335 BCE. Aristotle explains rhetoric as a systematic method of persuasion. He defines on the first page of his book "RHETORIC" as the counterpart of Dialectic [5]. Although he wrote his treatis *On Rhetoric* in the middle of the fourth century B.C., the interest aroused by his rhetoric has been more intensified in modern times than it ever was in antiquity or the Middle Ages.[6]

In his book [5], he gives a list of four different reasons why rhetoric is useful. According to Aristotle, (1) rhetoric increases likelihood that truth prevails over the lie, (2) use of rhetoric for a popular audience does not disadvantage people not instructed because it relies on notions possessed by everybody, furthermore (3) rhetoric reveals both opposites sides of an argument to determine clearly facts and finally (4) rhetoric is very useful to defend oneself with speech and reason, rather than with a physical fight.

Aristotle was the first person to recognize clearly rhetoric as an art of communication morally neutral. The neutral nature of the rhetoric implies that it could be used either for good or ill.

> Its persuasion, he says, depends on three things: the truth and logical validity of what is being argued; the speaker's success in conveying to the audience the perception that he or she can be trusted; and the emotions that a speaker is able to awaken in an audience to accept the views advanced and act in accordance with them. (*Aristotle on rhetoric : a Theory of Civic Discourse* [6])

In this quotation from the book of reference containing the chief work of Aristotle on rhetoric, he divided the means of persuasion, appeals, into three categories: *Logos, Ethos, Pathos*. These three categories of means are represented in the Fig 2.2 and described below.

The first one, *Logos*, refers to the logic, reasoning and method of construction of the argument. Much emphasis is placed on the internal consistency of the message, i.e. on the clarity of the claim, the logic of its reasons, and the effectiveness of its supporting evidence. This first category takes an interest in the rational mind of the speaker.

The second one, *Ethos* or ethical appeal, refers to the trustworthiness or credibility of the speaker. The current interest rates on the projection of the speaker's character, the style that he must take to capture the attention and win the trust of the audience, to make it credible and sympathetic.

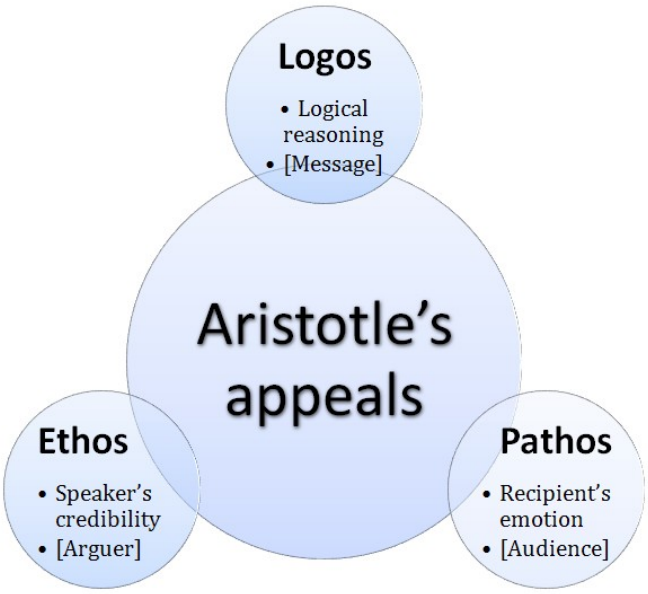


Figure 2.2: A general summary of Aristotle's appeals

Finally the last one, *Pathos* or emotional appeal, corresponds in awakening the emotions of the audience, persuading by appealing to the audience's emotions. It refers to the sensibility of the audience, not just to the emotional reaction but more to the fact to identify oneself to the speaker's point of view, to feel what he feels.

In addition to defining the three appeals of rhetoric, Aristotle speaks of three kinds of rhetoric, as described on the Fig 2.3. The styles of rhetoric are rigidly classified by Aristotle, and the use of one or the other depends on the occasion, the subject matter and the public. Actually there are three main areas for the public speaker: forensic, deliberative and epideictic. Each of these three kinds of oratory is largely motivated by the judgments required.

The representation given by the forensic oratory is a discourse pronounced by a lawyer who pleads a case in a court of law. Forensic oratory is legal arguments dealing with past events which must be proved and accusations which must be rejected. This kind of discourse is the most typical of the three and certainly the most central to rhetoric theory.

The deliberative type of discourse is relative to public affairs, such as economy or politics, and is generally delivered before an assembly which decides on these different kinds of public affairs. For example, it concerns political orators and their task to persuade the assembly to decide on a law, a declaration, etc. for future actions. Unlike the forensic genre, the aim for a deliberative type of discourse is to defend a future course of action.

The last genre of oratory is the epideictic one, also called the ceremonial oratory, which praises or censures somebody. The orators often find useful in this kind of discourse to discuss the present but also to recall facts or actions accomplished in the past by the person to judge and to make guesses at the future. For Aristotle, this genre represents nothing in the civic domain because it does not address

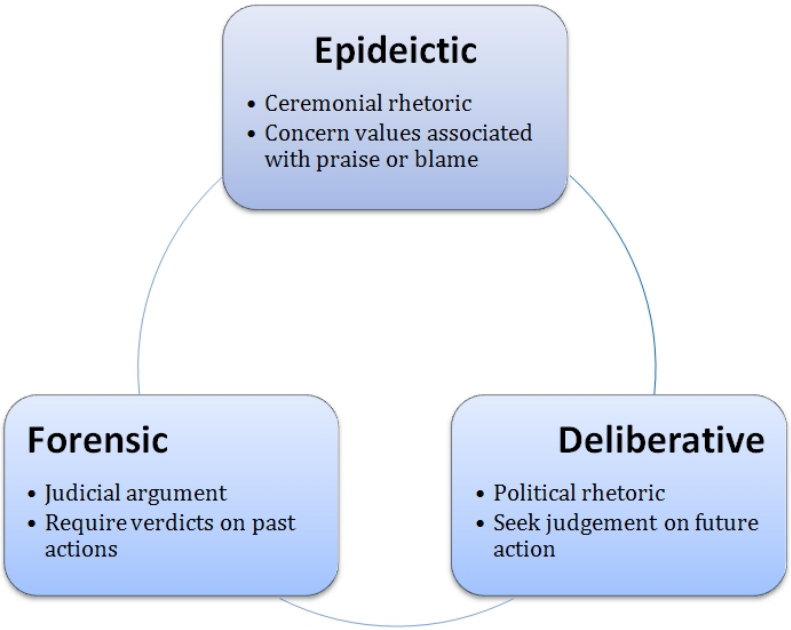


Figure 2.3: Aristotle's classification of speeches

the facts or policies but only values.

Through these different classifications, the work of Aristotle provided a solid foundation for the rhetoric theory. Although many modern theorists frequently refer to his work for the basis for rhetorical analysis or criticism of speeches and discursive writing, but also simply for ideas about communication [6].

Chaim Perelman is one among major modern theorists of rhetoric having been inspired by the work of Aristotle. He focused a long time on the third genre of Aristotle classification, the epideictic rhetoric concerned with values. This rhetoric was established without any standards for judging the content of the speech supposedly existed. It can stem from this that audiences had to be instructed to judge on matters of skill. The problem raised by Perelman sought the public's ability to judge matters of value. According to him, values were not judged on content but more on the style of the orator who uses them. This issue showed the need for a theory of argument in which values could be assessed rationally in order to be judged in the same way as facts and policies.

The first essential interest of the Perelmanian philosophy focus on the audience.[24] Perelman defines the audience *for the purposes of rhetoric, as the ensemble of those whom the speaker wishes to influence by his argumentation.*(See [36] p.19). Actually, the concept of audience defined by Perelman does not really refer to physical group of people assembled to hear the speech but only to the the virtual representation of the audience by the speaker or the speaker's mental conception of the audience. Then he divided the audience concept into two different part, the universal one and the particular audience. The former refers to the ideal audience, all reasonable and competent people, i.e. the incarnation of traditional reason, and the latter is some group of people whether or not they are reasonable or

competent. The conception of the universal audience is subject to valid conviction for every rational being, sometimes described as a mental construct varying from orator to orator, while the particular audience is subject to a particular instance of persuasion. (See [36] p. 28-29) Further information can be found in the fourth chapter of [15].

2.4.2 The Wigmore's Chart Method

In 1913, John Henry Wigmore, a U.S. jurist and expert in the law of evidence published an important book on the argumentation field concerning the legal evidences construction, entitled *"The Principle of Judicial Proof"*. In his essay, he specified rules for building a *Wigmore's "Chart Method"* to represent arguments with help of a tree diagram. This new visual representation aimed to bring a novel perspective in the field but also improve analysis of the mass of evidences demonstrated in legal cases. This practice tool will help advocates analyzing the proof of facts and judges conclude at trial. The Fig2.4 illustrates a part of the Wigmore chart which served in the case of Commonwealth versus Umilian in 1901.

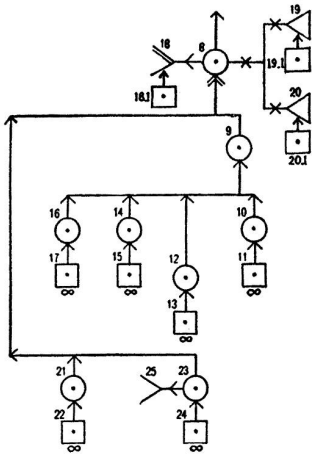


Figure 2.4: Example of the Wigmore Chart Method for analysing the evidence in a legal case.

Indeed, the Wigmore scheme is firstly a cognitive tool for reflection. He probably laid down there the first pillars of argument visualization. Nowadays, argument visualization can also be used in the legal context. For example, during a trial, each of the lawyers is building the diagram of the theory about the case, the theory which coincides with their client's interest. The visual presentation assists the judge concerned by the case to accept the most coherent legal theories which seem to be the most persuasive legal arguments [25]. Even today, the Wigmore's Chart Method is still taught in some law schools. The scheme is implemented as a basic scheme of argument mapping tools such as the Araucaria software which we will study more in detail later in this thesis. (See Section 5.4.1 for more information about Araucaria.)

2.5 The Dialectic Argument or the Conflict Resolution

The dialectic, as shown on the Fig 2.1 is another perspective of argumentation. Instead of rhetoric, the main goal of the dialectic is not to persuade the public of a claim, but to test the evidence by comparing its advantages and disadvantages. This exchange of arguments and counter-arguments leads to a critical assessment of the evidence allowing new knowledge, discussion, reasoning, questioning and interpretation.

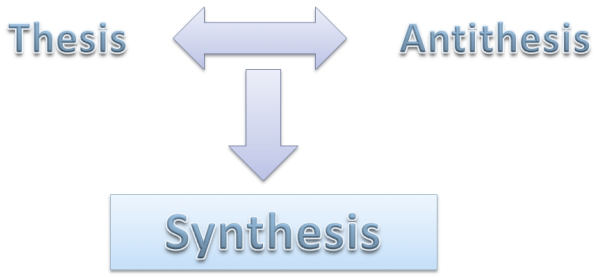


Figure 2.5: The dialectic principle

The principle of the dialectical argumentation is illustrated on the Fig 2.5. There are two forms of dialectic depending on the content of the synthesis, a strong and a weak form. The weak form results in a synthesis of the compromise between elements included in the thesis and antithesis, while the strong form goes beyond the agreement between ideas contained in both. The synthesis of the strong form provides new information corresponding to a further dialectical argument. A concrete application of the dialectic perspective is the Toulmin's structure explained in details below [46].

2.5.1 The Toulmin Context

For a better understanding of the real significance of the contribution of Toulmin's model and his ideas, it's essential to capture the context within which Stephen Toulmin, an English philosopher and logician, identified elements of a persuasive argument and by this way, carried out the main revolution in the theory of argumentation.

It happened in the 1950s, during the heyday of positivism, shortly before its collapse. At that time, positivists took the view that the physical world is based on logical inferences grounded in observable facts. They did not make any difference between sense perception and the external world. Only deductive and inductive reasoning were recognized as forming rational arguments. In this context, the Toulmin's book [48], entitled *The Uses of Argument* and published in 1958, introduced a deep adjustment by the consideration of the defeasible reasoning in the argumentation theory.

According to Toulmin, for judging of the soundness of an argument it is essential to determine two different elements. First, the value of the warrant which provides the link between data adduced in the argumentation with the claim, and second the assurance of the backing that the warrant is made acceptable.[58, 51]

It was really a challenge to the dominance in philosophy of formal, Aristotelian logic [8]. From

then, the argumentation issues were no more to know if the scientific law was true but if it was effectively applicable to the current case.

The Toulmin's Model of Argumentation

Here, we outlined precisely the structure of the Toulmin's Model of Argumentation and we provided an typical example to help the reader to understand correctly the different components of this structure of argument.

The Fig 2.6 illustrates the five components and the four relationships included in the Toulmin structure.

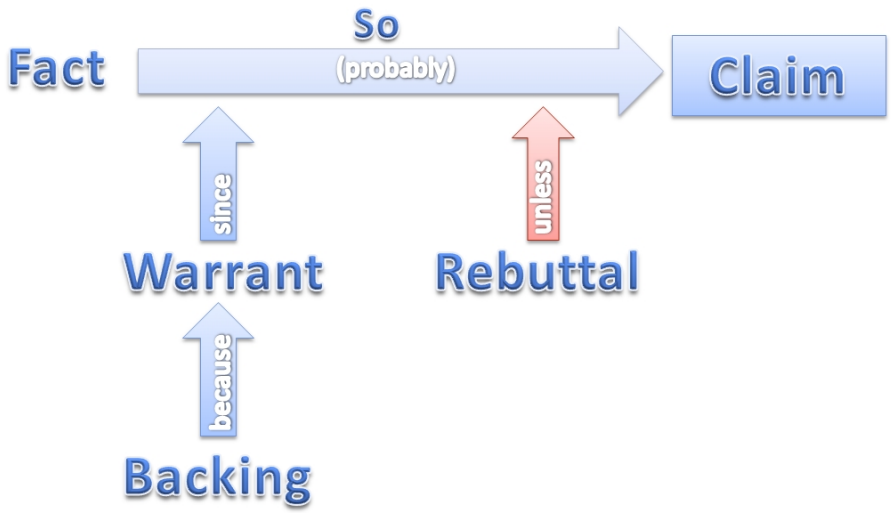


Figure 2.6: Toulmin's layout of arguments

Concretely, in addition to the concepts of *Claim* which is the statement to prove, and the *Grounds* which are data and hard facts, Toulmin introduces the concept of *Warrant* as a bridge or a reason legitimizing the claim by showing the grounds to be relevant. He defined the warrant as a hypothetical proposition that can be subject to defeat. In his model, the warrant is supported by a *Backing*, i.e. an assurance provided to support the inferential passage. He also introduced a degree of force, called *Qualifier*, related to the strength of the inferential connection. Finally, he assigned the name *Rebuttal* to the concept of exceptional conditions that might defeat the conclusion. Indeed, despite the careful construction of the argument, there may still be counter-arguments that can be used. The nature of a rebuttal and the antithesis in the dialectic theory is the same.

More than the definition of the six new components, it is a new model for the layout of arguments he suggested. Diagrams seem very different from natural language, these outline a particular spatial arrangement.

Example of Application

With all these defined concepts, we illustrate (on the Fig 2.7) now an example provided by Stephen Toulmin ([48], 104-5). Furthermore, some questions seems really useful to be able to find all the elements of the argument.

Claim: Harry is a British subject.

▷ *What are they basing that claim on ?*
Data: Harry was born in Bermuda.

▷ *Why does the fact that Harry was born in Bermuda lead they to believe that Harry is a British subject ?*
Warrant: A man born in Bermuda will generally be a British subject.

▷ *How can they make this claim ?*
Backing: The following statutes and other legal provisions : ...

▷ *How sure are they that Harry is a British subject ?*
Modality: The probability that Harry is a British subject is not estimated, but he is presumably.

▷ *If they are so confident that Harry is a British subject then why have they still some doubts ?*
Rebuttal: Because he has probably become a naturalized American. Anything could happen.

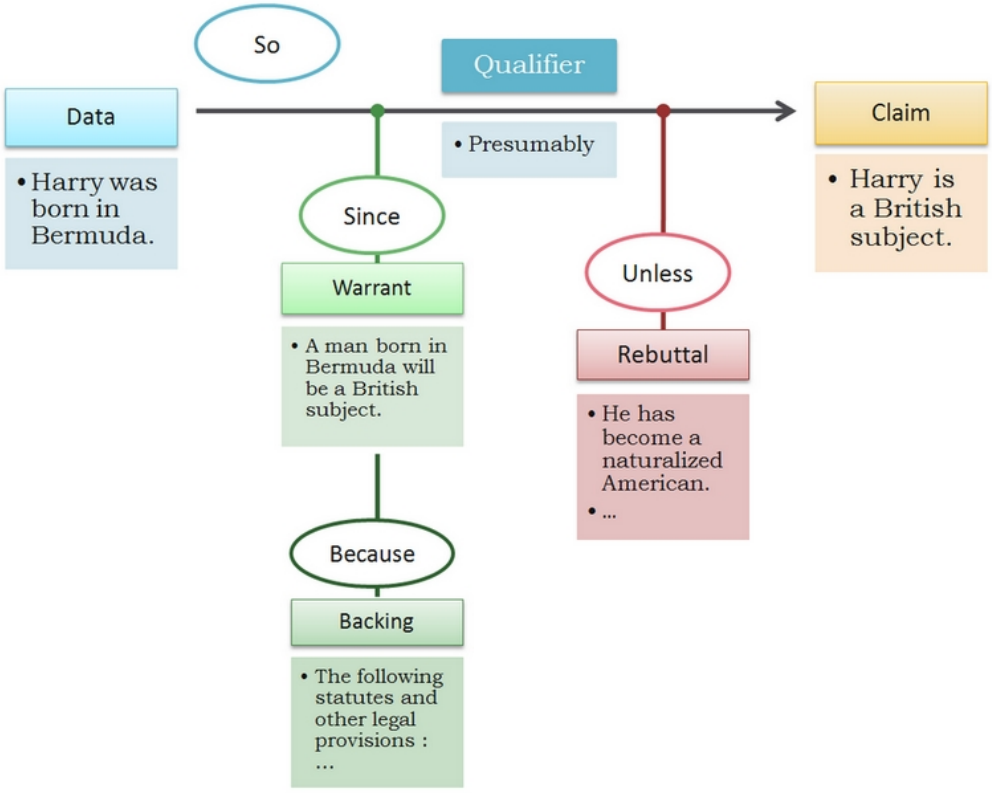


Figure 2.7: Harry example of the Toulmin argument structure [48]

2.6 Summary

In this chapter, we presented a quick review on argumentation. We first gave information about the argumentation discipline. Then we explained the three different topics of the foundation of the argumentation theory, the analytic argumentation concerned by the validity of the arguer discourse, the rhetoric argumentation used to reach the acceptance by an audience and finally the dialectic argumentation based on the critical assessment.

We also described the Wigmore's Chart Method and the Toulmin's Argumentation Model which are the first diagram types to represent a complete argumentation. These two different model appear as a bridge to the next chapter which focusses specifically on argument visualization.

Chapter 3

The Computer-Supported Argument Visualization

This chapter introduces the reader to the argument visualization, or more specifically to the basics and roots of Computer-Supported Argument Visualization[1] [2]. In particular this chapter puts an emphasis on the benefits of using these argumentation visualization tools to create, manipulate and read arguments in Section 3.2 and introduced some other concepts closely related to Computer-Supported Argument Visualization in Section 3.3.

3.1 Foundation of CSAV

The textual form of an argument is increasingly ignored for the benefit of its visual form because of the data visualization means are intended to be an aid to understanding. By Tim van Gelder [53], argument mapping is *diagramming the structure of argument, construed broadly to include any kind of argumentative activity such as reasoning, inferences, debates and cases.* Argument mapping is used to analyse the reasoning behind the derivation of an accurate conclusion supported by assertions and the constitution of evidential relationships between statements.

The argumentation process typically takes place as follows [10] : Pose or define problem, generate proposals, create supporting arguments, evaluate proposals and arguments, and finally make a decision, based generally on agreement or consensus. To support this process, CSAV provides the framework to help organize, display, and record this argumentation process. The result of this process produced with an argument mapping tool is a visual representation of all relevant reasons supporting or opposing the defined problem.

This representation is commonly named an argument map and is concerned with informal reasoning and concrete argumentation. It is typically represented by a *box and arrow* diagram. The diagram's boxes are corresponding to propositions, statements or assertions, and its arrows are expressing relationships among propositions such as logical, evidential or inferential support.

The example chosen to illustrate argument mapping appears on the Fig 3.1 and is excerpt from [53]. It has been created using the bCisive software, an easy-to-use software developed by Austhink,

[1] A deeper approach of CSAV roots can be found in [8].

[2] If the reader needs more information, he or she can also refer to [39].

for visualizing thinking, collaborating and improving workplace productivity. The map represents a portion of the argumentation found in an opinion piece written by Paul Krugman on the universal access to health care in the United States.

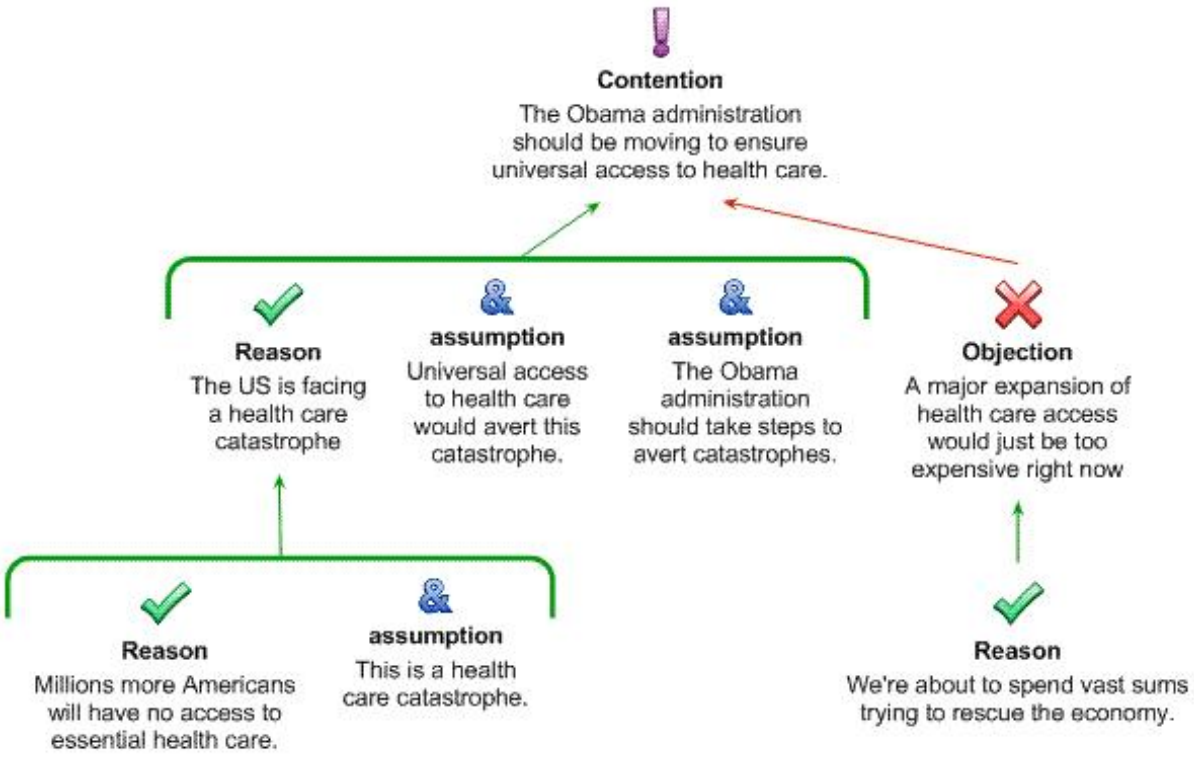


Figure 3.1: An example of an argument map achieved with bCisive.

The reader can see on this example an argument diagram in the form of a pyramid with, at the highest level, the contention or the problem to argue represented by a purple exclamation point with the textual information content. This claim is supported by a combination of three others statements, two assumptions and one reason in turn supported by a reason and an assumption. The claim is also opposed by an objection. For each different type of statement, clear icons, like the cross to express an objection or a vee for a good reason, and keen colours are used in the map, for their cognitive efficiency in the reader mind. The red color expresses a negative feeling while the green colour represents something good.

More formally, the argument map used as an example encompasses specific theory concepts of argumentation such as the contention object, assumptions, reasons or still the objection concept. As we seen the reader attention is directly attracted by the visual conventions, the colours widely used, lines, position in the map, labels, etc. All these visual variables help convey all the information necessary to the reader to understand and evaluate the current argument and its structure. A last approach the user can notice on this diagram is the aggregation of multiple supporting statements into one single construction to be the foundation of the upper level.

This example and its description leads to define precisely the three abstraction levels that are

distinguished for a mapping tool. In fact, two argument maps for a same argument could be very different following both the user intellectual ability and the software tool used. The intellectual ability of the user is a complex field not discussed here. By cons, mapping tools can be evaluated on basis of three different levels (Fig 3.2) :

Figure 3.2: The three levels of argument mapping information.

Argumentation Theory Level, such as informal logic, critical thinking, or rhetoric. The theory specifies the entities, relationships and values to be represented and provides rules or guidelines governing map construction.

Visual Conventions Level, for displaying arguments in accordance with the theory. There are not only dimensions important such as shape, color, or line, but also properties as communicative effectively, properly conveying to the reader the argument structure and associated issues; support interaction and please the eye.

Resources and Technology Level, for identifying technology limitations of models such as support for complex diagrams and their modification, and failure to constrain or guide the user in any way.

All these levels are going to be deeply studied for some argument mapping tools in the next chapter of this thesis.

3.2 Benefits of CSAV

3.2.1 Added-value for the Arguer

Before starting, we have to mention that arguers are people who represent arguments and create argument maps, a map can be created by one or more arguers.

The benefits of argument mapping for arguers are multiple. Firstly it promotes the clarity and the insight of more complex arguments that arguers have difficulties to represent totally in their minds or fail to explain easily what largely goes on in the head during a discussion between arguers. Indeed, an argument mapping tool is an invaluable tool for arguers to improve the communication about argument statements by making clear what claims are made and their relationships with others.

Furthermore, it gives a simple structure for developing a rigorous and complete articulation between the argument assumptions, the rebuttals and the conclusion. These argument maps are a way to conceptualize different elements leading to a whole argument. It could also help to enhance the reasoning skills, to find something missing from the one arguer mind, to correctly understand the point of view of other arguers and to finally make better decisions.

A main motivation behind much CSAV work is probably to augment the arguer intellectual ability in argument construction and analysis. [8]

3.2.2 Added-value for the Reader

Argument maps readers are people who attempt to understand correctly maps and evaluate the map content but also the arguer work such as it is the case for example with teachers who develop the critical thinking of their students by correcting their argument maps.

The benefit of an argument map for readers is that the stages of the arguer critical thinking are more obvious and easier to understand and to connect with personal knowledge. The level of abstraction is reduced because of the visual representation. The scaffold of ideas and reasoning is in many cases more difficult to understand under a textual form. The map provides obviously an important visual tool which can be monitored and assessed against criteria.

Among all CSAV tools, some are implemented with an evaluation function where the arguer can justify why he or she accepts a claim as true, or assigns it for example a such advised level of support. For the reader of this kind of work, the argument map is a valuable source of information.

The reader has sometimes the role of taking the decision to accept or reject the position presented in the argumentation. In this case, a better understanding of the map implies generally a better decision, even if the utilization of an argument mapping tool does not prevent a reasoning error.

3.2.3 In the Education Field

Articles devoted to the teaching of argumentation theory, informal logic or critical thinking are numerous, often included into critical reasoning or philosophy courses. Several authors assert ultimately that students improve their informal reasoning skills by learning the argumentation theory [49, 16].

However, teachers at the undergraduate level complain that students do not correctly parse arguments elements from a text. In fact, they recognize the main ideas of the text but are unable to follow the whole argument [20]. An improvement could be possible if courses were given by taking into account in addition to the theory, some practice exercises with argument visualization tools, although the results of studies conducted do not still prove it [10].

The learning of argumentation scheme such as the Toulmin's Model of Argumentation in order to identify and critique arguments' elements in practice could improve students critical thinking skills and their informal reasoning methods. Improve the intellect ability is an essential step for students in their education, because of these skills prepare them better to succeed in their life, in the world. The result can worth the trouble.

3.3 Related Concepts to CSAV

Computer-Supported Argumentation Visualization is a part of the human-centred technology which aims to bring human-centred approaches to the design of existing, new and emerging technologies for both work and leisure activities. Other concepts related to human-centred technology are quickly presented in this section such as mind mapping or brainstorming software, Computer-Supportive Coop-

erative Work, Computer-Supportive Collaborative Learning or Computer Mediated Communication.

The mind mapping concept was originally invented by Tony Buzan in the 1970s [9]. A mind map can be produced with a specific software in many different situations. It is mainly a tool used to represent the knowledge about a central idea, or to elicit information to solve a problem [28].

The process of the mind is represented by a diagram composed of words, keywords, lines, ideas, colours, images, sketches, or symbols. While an argument diagram is created from a claim to assess, requiring the consideration of supporting reasons and objections in the form of declarative sentences, a mind map deals with one main idea, linked with other items arranged intuitively around this central idea.

The mind mapping can be used to solve problems, to conduct a brainstorming session, to take some notes to better integrating knowledge, to memorize, to connect together different concepts, to give a good overview of important connected elements, etc.

An example of a mind map is given on Fig 3.3 to illustrate the concept. This example shows a comparison between four different tools of mind mapping.

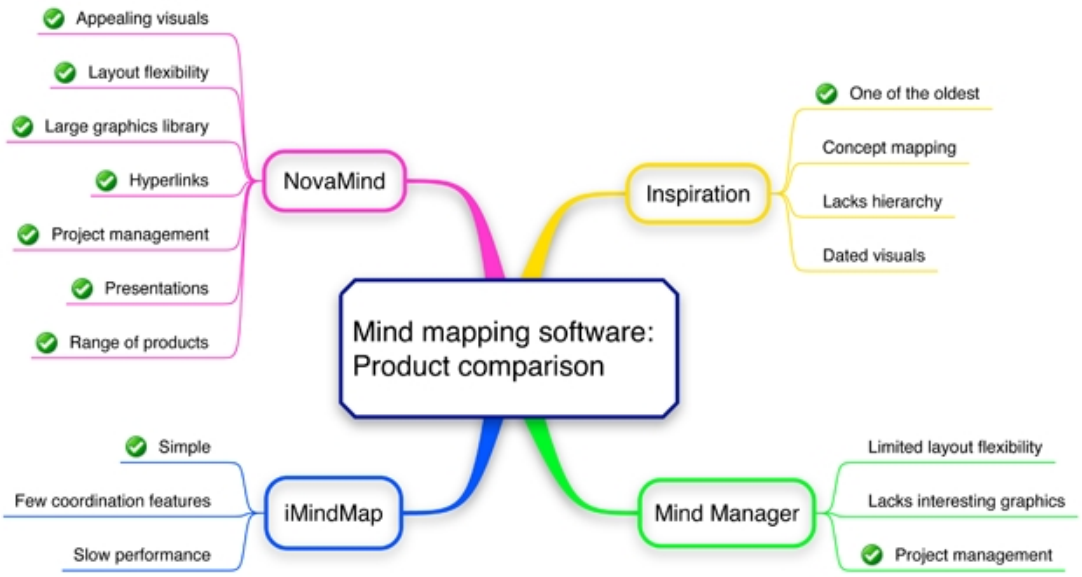


Figure 3.3: Example of a mind map comparing four different mind mapping tools.

Some names of popular mind mapping software applications are provided here for the reader information, such as FreeMind[3], Mindjet MindManager[4] and Buzan's iMindMap[5]. A collaborative system of mind mapping is also available at `http://www.mindmeister.com/`.

[3]FreeMind is open source and can be found at `http://freemind.sourceforge.net`
[4]Mindjet MindManager is a commercial software which can be found at `http://www.mindjet.com`
[5]iMindMap is a commercial software which can be found at `http://www.thinkbuzan.com/`

Mind mapping and brainstorming sessions can also be included in a global field called the Computer-Supportive Cooperative Work. The Computer-Supportive Cooperative Work is an emergent interdisciplinary field resulting in association of computing and social science. A clear definition of the real scope of this new field is not yet accepted by all, but a global perspective of this field, coined in 1984, aims to provide improvements of the production process gathering the work of multiple individuals in order to multiply productivity [29].

Videoconferencing software, application-sharing programs or software for tracking document changes are programs involved in CSCW. Networked computer systems are a new medium to overcome constraints such as physical or social abilities. This also applies to the collaborative learning supporting by computers and the Internet. Indeed computer-supported collaborative learning (CSCL) is another field enjoying of the same idea of collaboration work as CSCW but rather oriented towards the online education of users and their collaboration leading to the generation and circulation of new knowledge. The specific focus of this field is to assess how learners acquire knowledge and skills together [41].

The last concept which is introduced here is the computer-mediated communication defining any communicative transaction via computer-mediated formats (e.g., instant messages, e-mails, chat rooms). The field focusses mainly on social effects and the role of interpersonal interaction with this kind of communication occurrences. Human-human communication through computers and networks are one of the most important characteristics of this medium is the opportunities it offers for human-human communication through computers and networks [23].

3.4 Summary

In this chapter, we introduced Computer-Supported Argument Visualisation which is a specific field for organizing, displaying, and recording the argumentation process under a visual form, commonly called an argument map. We presented the main benefits of this human-centred technology and we introduced some related concepts to the reader.

As we already specified in the introduction chapter, this thesis is not only based on argumentation, but is also related with the computer security field. In the next chapter, we, therefore, presents the global context of security needs.

Chapter 4

The Security Requirements

> *In the United States, according to the U.S. Federal Bureau of Investigation (FBI) statistics released by the United States each year because of network security issues caused economic losses amounted to 7.5 billion U.S.*[62]

To survive in hard contexts of competition, companies know that is essential to protect suitably the most important business assets. Information is a valuable corporate asset. The threat sources to a company's information system are multiple and varied. Larger or wealthier businesses, and even small and medium companies can recognise security data management as a vital aspect for their business. Computer and information security is essential for the survival of the company because data on asset components are the first source of knowledge, skills and value creation in many businesses [34]. Protecting the system against natural disaster, fault and compromise behavior is a first step *"to ensure business continuity, minimize business damage and maximize return on investments and business opportunities"*[22].

This chapter is a large introduction to the "computer-supported" security. Firtsly, we present the issues and objectives of security in Section 4.1 and 4.2. After that Section 4.3 introduces the three main security requirements in IT systems : transport level privacy and encryption, authentication control and authorisation control. Section 4.4 introduces the Common Criteria standards used to define the level of security in a system. Finally, this chapter ends with Section 4.5, a security approach based on human and the organization of a business company.

4.1 The Importance of Security

The protection of information systems is a growing concern fuelled by the growth of widely distributed systems. Many surveys are interested in business security issues. Figures speak for themselves. $100,000 were spent in 32% of cases of directors surveyed for business interruption and disaster recovery. An huge amount which could certainly be avoided in many cases with a better security protection or could be minimized with a more relevant risk management. All results appear in the table 4.1.

Security warning and protection against various threat sources are once more an actual worldwide concern.
Security of information system refers to many substantial concepts the organisation has to preserve

Table 4.1: Results of a survey about business security issues [26]

Security trouble	% of those surveyed	Average annual loss
Business Interruption and Disaster Recovery	32%	$100,000
Data Breaches	32%	$50,000
External Theft and Vandalism	65%	$20,000
Fraud or White Collar Crime	37%	$11,000
Theft of Laptops	58%	$10,000
Employee Theft	68%	$10,000

such as the ethics by protecting privacy and individual data or the application of some deontology principles in gathering information.

4.2 Security Objectives

Information technology security requirements has been a study subject for decades of work. At the start of a project, security issues are increasingly integrated into the software development life cycle, mainly in the design phase. Security of information system need to be evaluated early in the process and periodically during the various others phases.

As seen in the course entitled *"INFO M115 – Sécurité et fiabilité des systèmes informatiques"*, security objectives are :

1. *Data Confidentiality*
   The property that information is not made available or disclosed to unauthorized individuals, entities, or processes (see ISO/IEC 7498-2). This first objective takes into account all data, including programs, while they are stored, processed or transferred. Data confidentiality is closely linked with the trust user have in the system.

2. *Data Integrity*
   The property that data has not been altered or destroyed in an unauthorized manner (see ISO/IEC 7498-2). This second objective takes also into account all data including programs, while they are stored, processed or transferred. To keep the system integrity, it requires defined rules of access with enforcement mechanisms. To restore integrity, system requires recovery mechanisms.

3. *Data Availability*
   The fact that services are operational and data is accessible in an acceptable response-time (see ISO/IEC 7498-2). Monitoring and alerts are a fast, convenient way to keep track of the system information and detect the early signs of troubles.

4.3 Common Security Requirements

The three data properties, confidentiality, integrity and availability, described in the previous section, are closely connected with security requirements expected in a secure system.

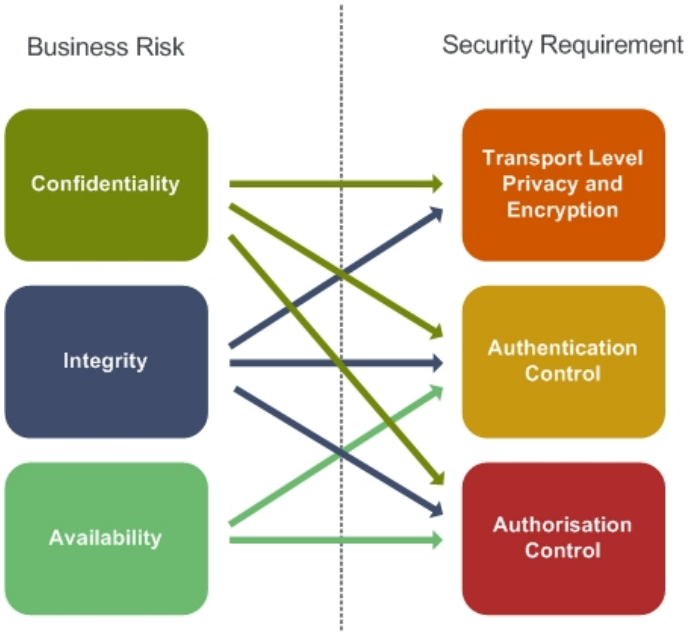


Figure 4.1: Mapping between business risk and security requirement

On the Figure 4.1 [1], the relationship between the global business risks and the common security requirements is illustrated to show the mapping between these key aspects of security and actions to enforce. This mapping is "common across all technologies and products"[**?**]. Transport level privacy and encryption, authentication control and authorisation control are three main security requirements in IT systems. These are described here.

- **Transport level privacy and encryption**

  Privacy is defined in ISO/IEC 7498-2 such as *"the right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed"*. The data transport level security and privacy is ensured by encryption mechanisms [45].

  Information exchanged in business communication are encrypted, i.e. transformed into an unreadable message for anyone without the knowledge of the key. Indeed, various cipher algorithms, such as AES or RC4, use a key during decrypting phase to recover the content. The key size can cause troubles in some cases and leads to weaknesses (for more information see [35] and [59]).

  Encryption is not only used during data transfers but can also be used on data stored in storage devices or on data in process. The process of encryption is used to preserve privacy and also

[1] Excerpted from `http://usefulfor.com/security/2008/07/09/middleware-and-me-part-2/`

other confidential business data. More than that, encryption process helps to preserve data integrity too. Encryption does not permit any modification without proving the knowledge of the key. The manipulation of encrypted data would result in defacing the original content.

- **Authentication control**

  An authentication control provides the identification of a person or a computer program before granting access to the system or a part of the system. It ensures that the individuals are who they purport to be, not impersonators. Password composed of a string of characters is an example of a confidential authentication information. Other mechanisms exist as the off-line authentication certificate binding an entity to a cryptographic key, certified by a trusted authority, or the other form of this certificate requiring an on-line interaction with the trust authority which delivered it.

  These authentication mechanisms identify and associate a person with a preregistered profile of user. Information systems with authentication control have an identity-based security policy, i.e. a policy based on the identities or attributes of users, a group of users, or entities allowing authorisation control to achieve some actions on the system. Identify a person or a machine via a secure mechanism of authentication is one among many good steps to manage correctly a system. The authentication control is followed by an authorisation control described into the next item.

- **Authorisation control**

  The authorisation control is a third security requirement complementing the two others in the establishment of a cross-layer security on the system. Authorisation process is closely related with the granting of rights. This includes therefore the granting of access based on access rights. Access control policy involves regular updating of sets of rules authorizing a group of users to perform a set of actions on a set of resources.

These three concepts are mainly used to express security requirements and to implement them. Indeed, their implementation and combination prove to be very useful to grant a security access to a system user. In addition, other properties are added by the ISO 13335 standard such as the authenticity, the accountability and the non-repudiation. All these concepts are the central interest for an organism in the design of security requirements for a system.

4.4 The Common Criteria

"The Common Criteria for Information Technology Security Evaluation" [60], called Common Criteria, is an international standard (ISO/IEC 15408) developed by six different countries through a combined effort to create a standard requirements catalogue for evaluating security of IT systems. Common criteria are also useful as establishing security requirements. *"The Common Criteria enable an objective evaluation to validate that a particular product or system satisfies a defined set of security*

requirements." [27]. Evaluation based on Common Criteria records the security claims and vulnerabilities level, i.e. a result on a scale of seven Evaluation Assurance Levels which can be compared.

Concretely, Common Criteria contain 11 classes including 60 security functional requirements. This common set of requirements are used to create two kinds of documents, a *Protection Profile* and a *Security Target*. The Protection Profile is a generic catalogue for specific consumer needs. For a class of Targets of Evaluation, it describes a set of implementation-independent security requirements. The Security Target is a refinement of the Protection Profile including detailed product-specific information and specific security requirements.

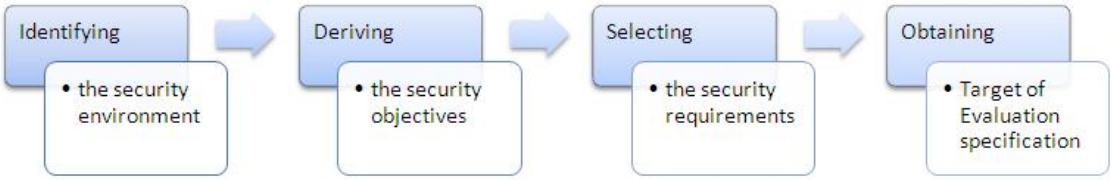


Figure 4.2: Common Criteria specification framework

The Fig 4.2 illustrates the Protection Profile or Security Target specification framework. It is built in three steps : identifying the security environment, specially threats, assumptions, organizational security policies; then deriving the security objectives, for instance the data protection or the system integrity. The last step to obtaining the Target of Evaluation specification is the selection of the specific security requirements which meet the objectives [60].

Security requirements are divided by their nature into two sets; the first one includes security functionality requirements, describing product features, and the second considers security assurance measure requirements, as illustrated on Fig 4.3.

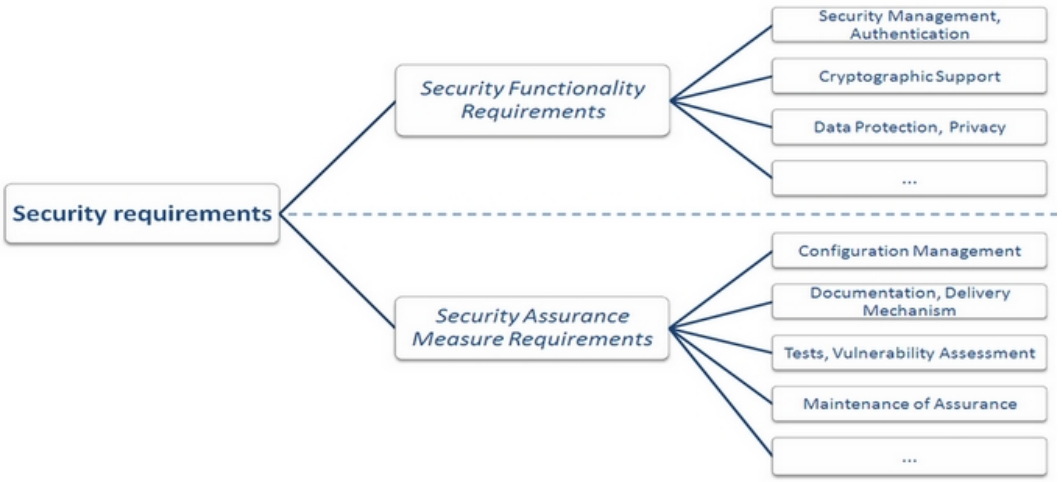


Figure 4.3: Common Criteria specification framework

The set of security functionality requirements describes many universal requirements such as security audit, non-repudiation, cryptographic support, user data protection, identification and authen-

tication, security management, privacy, protection of the target of evaluation security functions and self-protection, resource utilization limits or trusted path / channels definition.

Possible assurance requirements improve confidence others have in the system. This set of recommendations relates more specifically all the practices the system developers must have throughout the whole life cycle of the product development. Among these measures it is essential to notice the use of configuration management tools like configuration and version management system(typically CVS), suitable delivery and operation mechanism, consistent documentation and guidance documents, lifecycle support, tests and vulnerability assessment, and finally maintenance assurance for all change the software will undergo.

The result of this evaluation method of Common Criteria is summarized into an Evaluation Assurance Level, ranging from 1 to 7[2]. EAL7 is the highest level providing the more confidence in security functional requirement met in the system, while EAL1 describes a product with correct operation without any serious view for security threats.

4.5 Security Policy

In company context, the security officer is often face to the question : *How to determine the right level of security adequate for the organization ?* Based on the security requirement logic introduced a security policy defining the appropriate level of security. The security policy document *states in writing how the company plans to protect its physical and information technology assets* [3]. The policy *sets the boundaries of acceptability across the organization*[4].

The policy states a description of roles and responsibilities of those affected by and how the company's employees are educated to protect assets as follows their access level to the system. Defining clearly in the policy roles and responsibilities and what actions, activities and processes are allowed are the first steps to avoid any conflict with any user. Adding what the punishment will be for security non-compliance is also essential and mostly how security measurements will be carried out and enforced.

Getting the specific security policy document is not enough to be effective. Actually, in order to prevent fraudulent and unauthorized use of company's assets, the security policy must be deployed and accepted by all employees. Some essential properties of a security policy are : being understandable, realistic, consistent, enforceable, flexible. More than these basic properties, the security policy has to be documented, distributed and communicated properly to every employee to ensure efficient delivery and to keep people informed about what it is acceptable or not. This can be a tool to hold accountable people who do not comply with the rules.

Finally the security policy includes a procedure for evaluating the effectiveness of itself to ensure that necessary corrections will be made. Furthermore this should be often reviewed to prevent it becoming obsolete [43].

[2]Description of the seven levels can be found at `http://www.cygnacom.com/labs/cc_assurance_index/CCinHTML/PART3/PART36.HTM`.

[3]Definition from Whatis.com http://searchsecurity.techtarget.com/dictionary/definition/what-is-security-policy.html

[4]Information Security Policy World, http://www.information-security-policies-and-standards.com

4.5.1 Roles and Responsibilities

What is the role of information security in today's fast moving business environment ? As we have said, for many companies, this role is extremely vital. Technologies are developing very quickly and make the business vulnerable. Companies and their customers need secured communications to be able to benefit from the advancements. This is possible by implementing the adequate measures leading to an acceptable level of security competency. This section focuses on the security management in organization, more particularly the management inside an information system such as a company.

According to the organizational configurations model of Mintzberg (See in the course intitled *"INFO M421 Gestion stratégique des systèmes d'information"* and published in [30]), each organization can consist of a maximum of six basic parts described below and illustrated on Fig4.4[5].

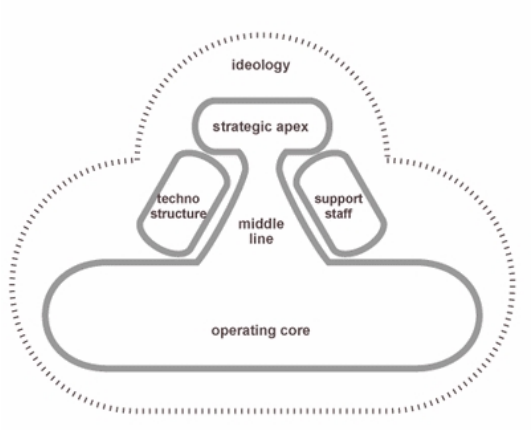


Figure 4.4: Six basic parts of the organization

1. **Strategic Apex** corresponds to the top management of the organization, where policies are defined.
2. **Middle Line** is a hierarchy of authorities between apex and core, permitting vertical and horizontal integration.
3. **Operating Core** is the basis of the organisation, where people do the basic work, the core business.
4. **Technostructure** is concerned with adaptation to change, environmental scanning and work performance.
5. **Support Staff** appears as an indirect support of the operating core, of the standardised work.
6. **Ideology** acts as the *glue* that holds collective actors together with beliefs and traditions; norms, values, culture.

Security policy is included in the ideology of the company. Security is nested in every part of the organization. Security is not only the interest of the security manager of the company. The security management starts by the strategic apex who defines the security strategy by means of security policy in a comprehensive way for the company.

[5]Excerpted from `http://www.provenmodels.com/22`

Then, the security IT strategy is defined by the technostructure, mainly by the security manager, aligned with the strategy. Normally, the security policy defines clearly what actions, activities and processes are allowed and which are not, so business functions align with them depending on what is technically possible to do. Every security functional requirement is tested in all business functions. Technology alone is not able to implement completely all security requirements. If some actions or activities made by an employee does not meet security policy, it is important to have defined the punishment for security policy non-compliance.

The role of the security manager is to evaluate the acceptable level of risks to cover. A security manager has to provide assurance the information security strategies are aligned with the business goals and objectives and furthermore has to be consistent with applicable laws and regulations. So, to better manage the security requirements of the company, the security manager has to have a good knowledge of business objectives and information security concepts, but also of the relationship between information security and business functions. Other knowledge is essential for the security manager as the accepted international standards for information security management, the centralized and distributed methods of coordinating information security activities and methods for establishing reporting and communication throughout his organization.

All employees need to know the security policy, to know exactly what is secure and what is not. Security is more than just a software or hardware problem. It concerns also the communication network with others stakeholders, such as customers and suppliers, the human context of the organisation and the staff. Another big importance risk manager could meet, is to convince all the people who are working for the organisation to have the same idea of what is the security needs. Many functions of authentication or others may seem useless or needless for a simple member of the staff but the protection of data has many stakes to always treat with care.

Finally, it is also the responsibility of all employees of the company to respect the internal security policy. Security troubles are no more only related to the software system implemented on company hardware but also to all the information system including the communication system [17]. The security policy has to be applied correctly by all employees of the organization, and employees who do not comply with the security policy in the whole company will face penalties.

4.6 Summary

In this chapter, we presented a large introduction to computer security covering security objectives, specific security requirements, standards of security and their evaluation, and finally security in business with the presentation of the security policy issue.

We conclude here the background part of this thesis, before we proceed to investigate the assessment, in the next part of this thesis.

Part II

Contribution

Chapter 5

Overview of the Assessment Process

With the emergence of computer science, public interest explores this new potential to create software packages of argument mapping tools, for solving most of complex problems. These frameworks provide new abilities for users, particularly for teachers and students, allowing them to customize argument representations. Computational features such as data confrontation or automated evaluation of argument premises are more and more implemented. The interest of developers for such tools has increased for more than a decade and a wide range of these new applications expand in many areas. Nowadays, the propagation of this kind of software is such that conducting an Internet research on the different mapping tools is quickly successful. Among those, some of the more prominent are Araucaria (5.4.1), Argumentative (5.4.4), and Rationale (5.4.5) to discover in the section 5.4.

This contribution part appears in this context with the objectives of **carrying out a review of a selection of tools used for argument mapping** and **determining keys criteria, functionality and qualities of an argument mapping tool that best satisfy the design of a security requirements argument**. These two main objectives are both the prevalent ideas of the whole evaluation process described along the contribution part. The relationship between all these tools building argument diagrams and the design of security requirements is probably not clear directly. Nevertheless, the interest for us at the end of this assessment is to list the key features and key properties expected by an argument mapping tool to represent security requirements. Argument components of security requirements are very similar to other more traditional arguments presented in philosophical or legal texts. This analysis will be therefore only very slightly oriented information security requirements development.

This chapter is the introduction of assessment process highlighting assumptions. It will be followed by two other chapters, one describing each of the tools results, and other summarizing the evaluation process and starting a discussion about the abilities provides by these tools. The plan of this first chapter is as follows. The section 5.1 explains the approach used to provide ratings of diagramming tools. It answers the following question: What are the pillars of the evaluation and the required steps to build this assessment? Next, the section 5.3 describes the user profile selected following his cognitive abilities and his personality traits. An overview of the argument of security requirements to design with tools is then described in the section 5.2, before giving to the reader, in the section 5.4, the selection list of argument mapping tools. Finally, Section 5.5 provides the assessment criteria.

5.1 The Assessment Methodology

The methodology process was developed on basis of a list of questions related to both main goals, such as what are objectives that argument diagramming software packages pursue, how to do these software's work, what type of user is targeted by each tool, or, what are the salient features implemented into the packages. The assessment answers clearly some relevant of these questions. This first section explains the overall approach of our methodology for developing new knowledge and skills in setting up this assessment.

The overall process is to select a type of user, to give him a specific task to perform with all tools selected for the assessment and to ask him finally to evaluate the application of these tools on basis of defined key criteria. Activities that go before result data collection are illustrated by the Fig 5.1.

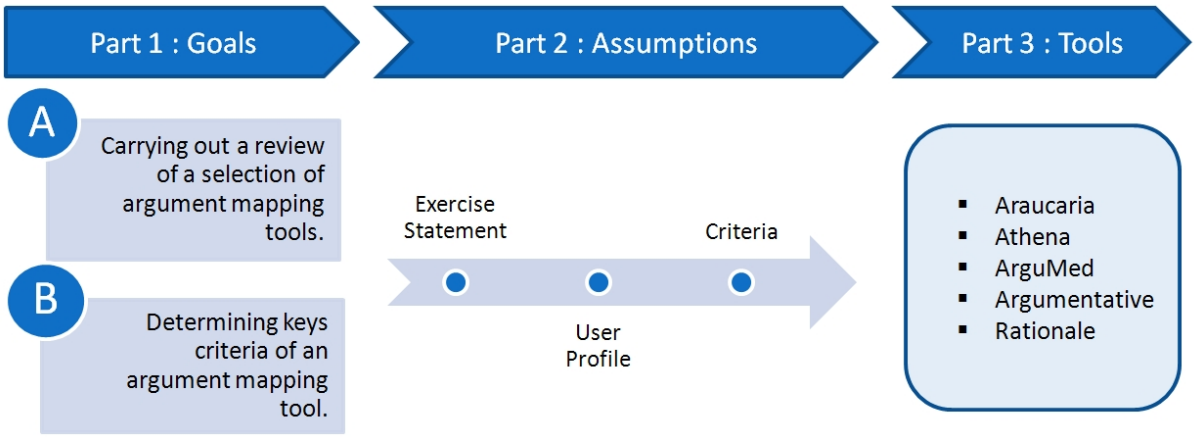


Figure 5.1: Schema of the methodology used in the design of the evaluation.

· **Step 1. Formalizing correctly assessment objectives according to our current needs.**
The primary need is an overall knowledge of existing argument mapping tools and the second is getting a list of relevant key characteristics to build security arguments argument. Therefore, the first goal is to browse features and possibilities of the most relevant tools and analyse them to produce a complete list of relevant features and properties applying to security requirements tool elaboration.

· **Step 2. Defining the task, the user profile and criteria for a compliant assessment.**
Our need to match the argumentation theory with the field of security requirements led us to define an argumentation exercise completely oriented toward the security conception and assign this task to a particular user profile. A relevant choice among standard criteria helped to establish a qualitative and functional analysis of tools.

· **Step 3. Selecting tools and compiling a complete record for each tool.**
The next step was the necessity to select and describe accurately the appropriate argument mapping software on basis of their description, their scope or their reputation.

After all these specifications, the constraint was then to place oneself in the shoes of the user corresponding to the profile to get a handle on the different tools with the objective of being able to perform the task with. This familiarization with software interface and features allowed to analyse them correctly and determine their limitations.

5.2 The Exercise Statement to Complete

Let's look firstly at the nature of the task to achieve or, commonly named the assignment. The selected exercise to achieve is a simple problem of authentication (excerpted from [1]), the process by which you verify that someone is who they claim they are in a IT system. The feature used here is the password authentication. This way is commonly used in systems to manage access control to resources. This authentication method is not always the most efficient but can be totally sufficient in many systems. There exist various other methods of demonstrating identity, such as a smart card, retina scan, voice recognition, or fingerprints, but this simple exercise will only focus on the password authentication. Let's discuss arguments for the user authentication trustworthiness.

The first objective to complete is to define correctly the claim to support. In this case, the claim is that **the user authentication protocol of the system is secure**. No need to define exactly what the system is or what are the technical details of the authentication protocol because we are working for the moment at a higher level of specification. Main elements which support this claim are firstly **the password authentication is presumably secure** and **the user authenticates by supplying username and password**. This basic argument is expressed in the graphical form at the Fig 5.2. The Toulmin structure was adopted to represent this first form of the argument (See Section 2.5.1 for the Toulmin structure).

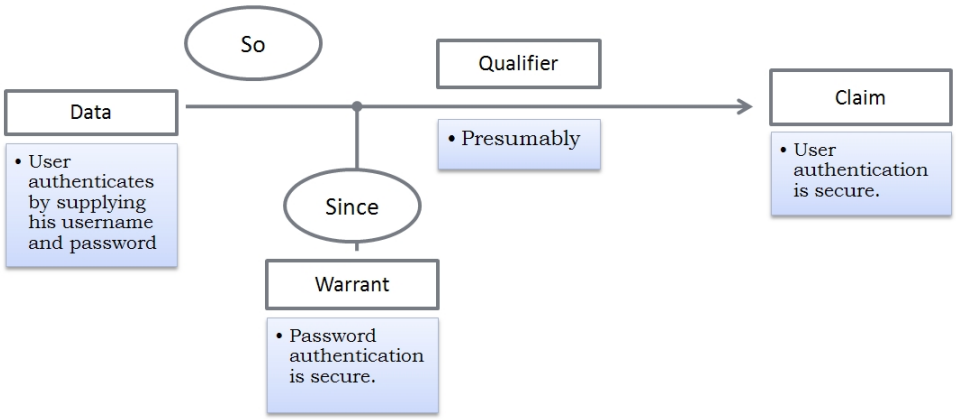


Figure 5.2: Form 1 of the password authentication exercise

The authentication function reacts in two ways, either the username and the password are correct and then the application grants access to the specific user based upon his credentials in the system, or else the password or the username or the pair is wrong, leading to an authentication failure. In failure cases, the system throws the specific error to the user and return him to the authentication form.

Threats lurking in this simple system related with bypass security mechanisms can be numerous. Chosen attacks of passwords for this study are the dictionary and the brute force attack which are the most accessible methods to compromise user's authentication credentials. The dictionary attack consists of trying variations of passwords found in a dictionary entry. If the dictionary attack is not effective, malicious user can try a brute force attack consisting in successive searching of all possible combinations. An account harvesting vulnerabilities in the application allowing to collect all the legitimate account names on the system from information provided in error messages is also studied.

The starting argument is now represented on the Fig5.3 by taking into account the threats under the form of Toulmin rebuttals.

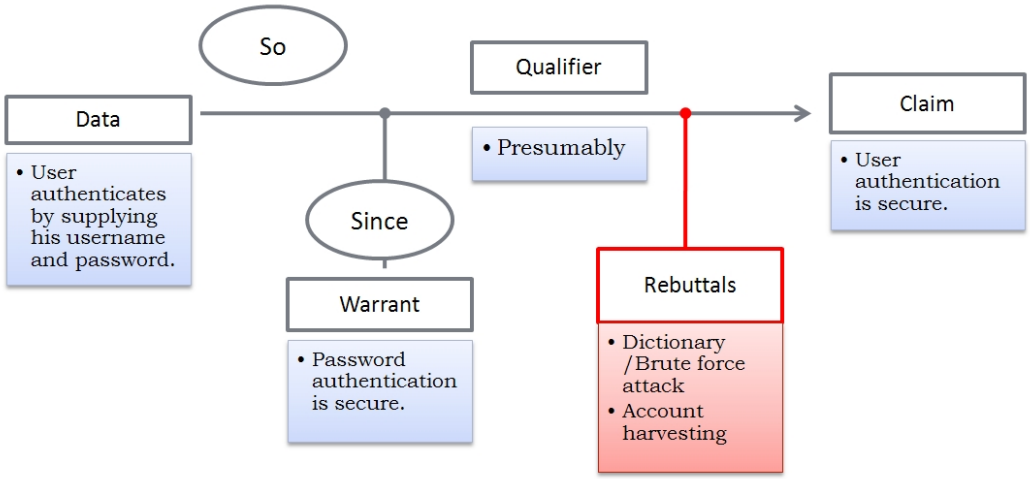


Figure 5.3: Form 2 of the password authentication exercise with rebuttals

These threats previously explained, are affordable and easy to implement. It is therefore essential to plan some measures to protect the system against this kind of threats during the security requirements design. One of the first measure to take on passwords concerns their minimum length and their complexity. Some policy controls on passwords can be imposed to users in order to achieve a sufficient security level. Another requirement is putting a limited number of attempts at entering a correct password before locking the account. Furthermore, in order to avoid account harvesting, developers must pay attention to not reveal relevant information in error messages, these need to be very generic.

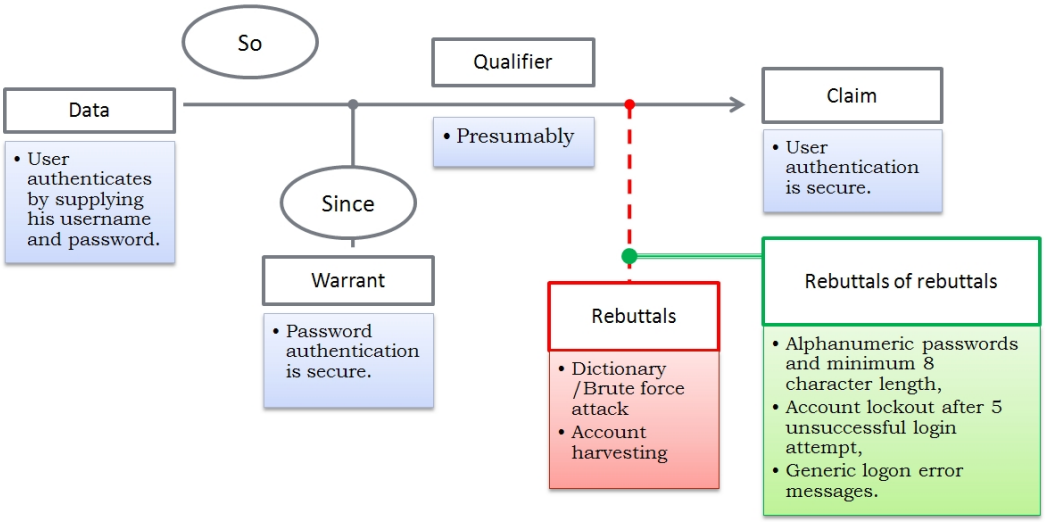


Figure 5.4: Form 3 of the password authentication exercise with rebuttals of rebuttals

The Fig 5.4 illustrates the idea of the rebuttal of a previous rebuttal with the adding of these

security requirements. Another illustration could have been to separate every rebuttals and to connect these which are disagree between them. The way that each tool allows to represent counterarguments is one of our preoccupation in the assessment. Finally, the user will have in front of him the argument as described on Fig 5.4 with notations and scheme implemented in every tool.

5.3 The User Profile

To best determine the *target audience* of argument mapping applications, user profiles of every tool need to be standardized and very generic to not dealing with system specifications. The general checklist for an evaluation includes four major categories of user characteristics: cognitive, personality, social, and physical. Obviously, any given software tool cannot be designed for every individual difference. In this specific context, only the study of the cognitive abilities is relevant.

A definition of the cognitive abilities could be the brain-based skills and mental processes that are needed to carry out any task. The identification of information about the user is defined in terms of his aptitude in his own domain of knowledge, his technological aptitude, and his knowledge of the argumentation theory.

Domain Expertise The task to achieve is inherent in the user's own domain of knowledge. In this case, user will apply his expertise regarding definition of security requirements. A security expert has all the specific abilities to work out the content of arguments and ensure their accuracy and completeness.

Technology Literacy The second domain is focusing on the computer skills of the user. The technology literacy is more than simply computer usage. Thomas and Knezek [47] defined it in including: *demystifying technology through conceptual understandings of the underlying science and mathematics principles, operational competence with modern technology systems; the ability to evaluate and use a variety of common technology applications, etc.*

Argumentation Knowledge This domain gathers user knowledge base, expertise and experience with argumentation theories. Knowing the multiple argumentation theory models and complex behaviour of inference rules or relationships between premises, rebuttals and conclusion may be useful in the use of such software tools.

The profile of users responding to the needs of this analysis is shown in the Fig 5.5 as experts in their domain and computer competent, but are not leading experts in argumentation.

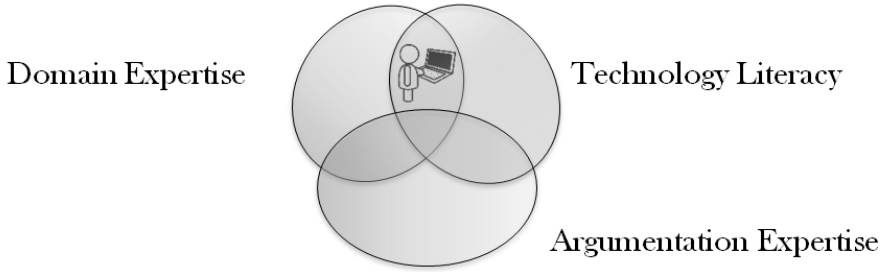


Figure 5.5: The selected profile for the user

The choice for this particular user profile can be explained by some reasons.

- Firstly, this profile may allow reaching a wider audience of users.
- The fact that we are not considering argumentation knowledge of users can be covered by the intention to evaluate these tools for their ease of use by novice and the simplicity of the task to perform, i.e. we distinguish the content brought by the user and the form obtained by the software tool.
- As the user is an expert in his domain, he is fully aware of the theoretical describing the system behaviour he wants to represent by visual arguments.

5.4 The Tools Description

Hereafter, the reader will find the list of the seven tools selected to be evaluated.

Both first tools, *Araucaria* and *Athena Standard*, were designed specifically to support the teaching of philosophy students. Their main objectives are to improve the student's ability to follow arguments presented in primary source texts and to develop the student's critical thinking.

ArguMed is an argument diagramming systems developed for a specific utilization in the legal field. This assists user, specifically lawyers or in law students, to create and understand models of legal argument. This particular orientation into the legal field still allows to design more traditional argument.

Argumentative system is also a free software not directly assign to a particular field. It allows to define different kinds of arguments from the simple scheme it suggests.

The assessment would not have been complete enough without the assessment of a commercial service. *Rationale by Austhink* can be used for learning and teaching in schools, but also in professional contexts to be able to resolve conflicts or to undertake discussions to obtain some consensus with mapping exercise.

These five tools seem to be very different a priori, nevertheless this assessment will prove they enable all of them to build a security requirement argument as defined in the previous part. Let's enter now in a deep description of every tool selected in order to better identify them and their advantages.

5.4.1 Araucaria

Figure 5.6: Araucaria 3.1 presentation logo.

Developers of Araucaria give a first description of their tool by saying : *"Araucaria allows you to build argument diagrams by selecting phrases of text to represent the premises and conclusion of an argument and building a tree diagram interactively using the mouse.*[1]. Used especially in the field of education, Araucaria seems to be a useful tool for learning and teaching the argumentation theory. This tool was selected to perform services to complete this assessment for its notability and recognition in the field. Before to immerse oneself deeply in the study of this tool, let's look at an overview describing the Araucaria's key characteristics, with its strengths and weaknesses.

Details : Araucaria 3.1 (2006) – GPL – Windows, Linux, Mac
Gleen Rowe & Chris Reed – University of Dundee (Scotland)

Objective : Preparing teaching materials in critical thinking, informal logic and argumentation theory for working with argumentation schemes.

Type : Argument-assistance system

Audience : Education, Research

Download : `http://araucaria.computing.dundee.ac.uk/doku.php?id=version_3.1`
See [38, 37, 42] for documentation

Strengths

· Supporting different already defined argumentation scheme sets (such as Toulmin, Wigmore,...).
· Providing a user-customisable set of schemes with which to analyse arguments ;
· User argument scheme sets are coherently integrated in the tool ;
· Providing an interface connected to the Araucaria database on-line repository of marked up arguments (request a registration) ;

Weaknesses

· Argument diagram building is almost inevitably on the basis of a text file.
· Lack of a basis tutorial for understanding how to construct the first diagram ;
· The position of boxes and arrows in the diagram is definite and unchangeable by the user ;
· Problems encountered with image saving.

Other

· Tutor mode available to correct students' diagrams.
· Colour background customizable.
· Developed in Java to support execution on many platforms.

[1] See in the Help file of the tool.

5.4.2 Athena

Figure 5.7: Athena 2.7 presentation logo.

The Athena software consists of two software packages along with educational modules. The first one, Athena Standard, is designed to support reasoning and argumentation. And the second, Athena Negotiator, is designed to facilitate analysis of decisions and two party negotiations. It should be noted that this tool is designed to teach before everything else. As many teachers will tell you, it is very hard to teach such inner mental skills as reasoning. The security requirements elaboration is also a complex reasoning task on which it is easier to argument when the issue is externalized. Only the standard module was evaluated in this assessment because Athena Standard is another tool among software packages available on the market that facilitate argument reasoning.

Details : Athena 2.7 (2008) – Free use for non-commercial purposes – Windows, Linux, Mac
Bertil Rolf - Blekinge Institute of Technology
& Charlotte Magnusson - Lund Institute of Technology (Sweden)
Objectives : Improving critical thinking skills.

Type : Argument-assistance and evaluation system
Audience : Education, Research
Download : `http://www.athenasoft.org/sub/software.htm`
A tutorial is included in the help section of the software - See [40] for documentation

Strengths

· Intuitive point-and-click and drag-and-drop user interface is simple enough for even novice to control argument diagram.
· Possibility to open several projects in the same time and arrange projects windows in several different ways.
· The generation of an Athena report allows to choose between several types of reports and takes into account user reflections on crucial issues of the argument.
· Evaluation of the trees step by step following acceptability and relevance rates.

Weaknesses

· The lack of an information bar is really a disappointing aspect of the tool because any other mean is not used to replace it.
· No assisted positioning function is implemented to display the argument with a better rendering.
· Poor argument scheme

Other

· The Athena software is the outcome of a joint project between philosophy and engineering science.
· It has been tested on some 160 students with favourable outcome.

5.4.3 ArguMed

Figure 5.8: ArguMed 3 based on DefLog presentation logo.

The ArguMed-system is an example of system for *computer-mediated defeasible argumentation*[2]. After the development of the Argue!-system in 1998, the ArguMed-system has been designed to enhance the familiarity of the interface and the transparency of the underlying argumentation theory of its precursor [55]. Although it is specially designed for lawyers, the interest found in ArguMed for this assessment was its property to be template-based, the user gradually constructs arguments by filling in templates that correspond to common argument patterns, and also allowing free argumentation *in the sense that it allows not only inference (i.e., 'forward' argumentation, drawing conclusions), but also justification (i.e., 'backward' argumentation, adducing reasons).*[54].

Details : ArguMed 3 based on DefLog (2001) – GPL – Windows
Dr Bart Verheij - University of Groningen (The Nederlands)
Objective : Supporting the reasoning process of the user

Type : Argument-assistance and evaluation system
Audience : Law
Download : `http://www.ai.rug.nl/~verheij/aaa/argumed3.htm`
See [57, 56] for documentation.

Strengths

· The interface is underlain by CumulA, a procedural model of defeasible argumentation.
· Keeping track of the issues that are raised and the assumptions that are made.
· Keeping track of the reasons and the counterarguments adduced, the conclusions drawn.
· Evaluating the justification status of the statements made.
· Logic-based evaluation in the background.

Weaknesses

· 3 panels available but changes can only be made on the dialectical argument panel to be updated on others panels.
· No drag-and-drop feature to create a relationship between statements.
· The predefined argument scheme is very simple, no use of qualifier or weight.
· No link to a help file include in this software tool.

Other

· The software is easy to use, so no really need of a tutorial.
· The archive of the software includes many examples for improving the user experience.
· Arguments diagram represented by an automated tree of statements.

[2]expression used mainly by B. Verheij to mean computer supported argumentation system

5.4.4 Argumentative

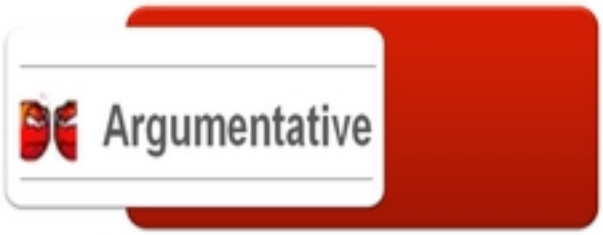


Figure 5.9: Argumentative presentation logo.

Argumentative system let user built clear argument maps with thesis and counter thesis. Thanks to its clear way of displaying arguments and objections and its group of relevant icons in the toolbar, the usability and the user experience are mostly right. Argumentative seems to be an indispensable tool to prepare debaters to argue. It is exactly the kind of tool sought to participate in the assessment.

Details : Argumentative 0.5.55 (2008) – GPL – Windows
John Hartley
Objective : To create, translate and manipulate argument maps.

Type : Argument-assistance system
Audience : Education, End Users/Desktop, Legal Industry, Religion
Download : `http://sourceforge.net/projects/argumentative/`
See the Argumentative manual [2] for documentation

Strengths

· The elements may be dragged and dropped under other elements to form the appropriate structure.
· The Argumentative system allows to export argument map to Microsoft Word and Power Point.
· The user interface offers an enhanced user experience.
· The graphical view was studied to meet much as possible the user needs.
· The tool includes a spelling check feature to correct the misspelled syllables.
· Loading several argument file formats (Araucaria or Rationale styles).

Weaknesses

· The software is fraught with many display and functional errors.
· Updates of content are not always directly passed in the graphical view.

Other

· Argumentative is intended to compliment mind mapping software with the ability to copy and paste between FreeMind and MindManager.

5.4.5 Rationale

Figure 5.10: Rationale 2 presentation logo.

Austhink, in collaboration with the University of Melbourne, has undertaken extensive research to determine the impact of argument mapping for critical thinking development on basis of courses undertaken with *Reason!Able*. Developed from Reason!Able, Rationale offers big improvements over its younger sibling. The choice to include Rationale 2 in the assessment list of tools is determined by the expected enrichment obtained by a comparison between free tools and a commercial tool. Indeed, a commercial tool should provide a software application of high quality with all the necessary documentation available. A complete trial version of the Rationale system was available during 7 days. All tests were realized during this short period.

Details : Rationale 2.0.4 (2008) – Commercial License – Windows
Publisher : Austhink Software -

Objective : Rationale helps improve thinking by providing an easy way to diagram reasoning on any topic.

Type : Argument-assistance and evaluation system

Audience : Education, law and business

Download : `http://rationale.austhink.com/download`
See [3] for a quick start with Rationale 2

Strengths

· Rationale has four map formats, including reasoning and advanced reasoning maps that create argument maps.
· An overview window shows what is outside the visible workspace.
· Rationale permits to Drag text and images from the web.
· Possibility to collapse or reveal some branches of the diagram to focus attention on particular lines of reasoning.

Weaknesses

· Price

Other

· The local and online information and help systems are very efficient.
· Numerous resources are available on the website[3] for learning and teaching in schools, colleges and universities.
· Rationale has the special feature to display a background image behind the argument map.

5.5 The Description of the Assessment Criteria

In order to achieve the most complete assessment possible, every tool is evaluated under five different approaches. These approaches gathered in two different points of convergence as described on the Fig 5.11: the level of user experience with each tool and the key characteristics of the argument diagram. The former appraises the contribution of these different applications to make possible for user to express and work with argument. This part of the assessment tests the intuitiveness of the user interface, the features and computational services available and documentation abilities suggested to the user to read and write arguments with each new tool in a manner appropriate to the context. This last properties is called the tool proficiency. The latter point of convergence examines all explicit constructs forming the argument's ontology inside each tool and the selected notations for rendering it.

Both these focal points are complementary and relevant because both allow to complete the as-

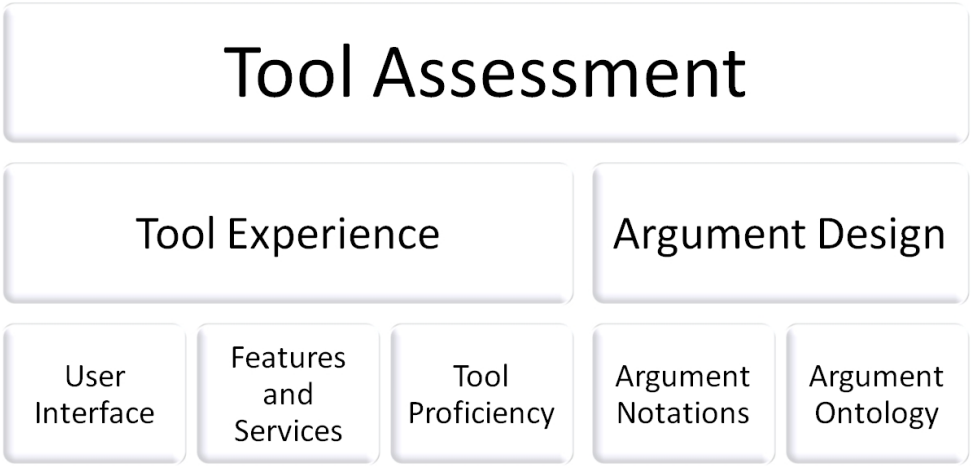


Figure 5.11: Assessment Structuring

sessment of each tool in its entirety. Indeed, the assessment covers specific criteria from the most tangible aspect of the tool, e.g. the completeness of the argument schemes underlying the tool, to the most abstract point of comparison related to the usability of the tool. Furthermore, identifying and laying down evaluation sub-domain for both approaches aims to determine a coherent evaluation grid [50, 53]. The subdivision also helps to reach a better demarcation between all these concepts that provides therefore a deeper understanding of the assessment process. Finally, the assessment provides a quality review of the tools.

5.5.1 Tool Experience

The first part of the assessment focuses specifically on the user experience during the tool utilization. User experience is about the feeling a specific user is affected by, while he uses the tool in a right way. The difficulty of expressing emotional feelings led the assessment to test concrete tools details in order to establish from this way a relevant awareness of the great potential of each tool. The detail on which the assessment focus are the user interface, the functionality and documentation and errors met during right utilization of each tool.

Let's describe more deeply these different tool components.

User Interface Convenience

Many techniques achieve user interface evaluation, as expert review, user review or interactive usability testing. The ultimate objective of the assessment is not to challenge deeply user interface components, but rather to assess the impression of three relevant criteria for user interface : the *understandability*, the *ease of use* and the *attractiveness*.

The understandability is the first criteria evaluated in this part, it is defined as *the capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use*[21]. This criterion is hard to test in a simple manner, so the assessment focus rather on the misunderstand-ability concept which related to the different panels of the user interface.

Table 5.1: Questions about *User Interface Convenience*.

Criteria	Questions
Information Accuracy	How accurate is the information provided to the user (insufficient, good, too complicated) ?
Balanced Interface	Is the workspace correctly shared between panels ?
Interface Intuitiveness	Is the user operating interface intuitive and simple enough for users ?
Visual Rendering	Is the argument presentation assisted by software to get a better visual rendering ?
Visual Attractiveness	Is the interface appealing to users ?
Diagram Scalability	Is the software tool designed to conceive large scale arguments ?
Direct Manipulation	Does the interface allow the modification of argument maps ?
Zoom Magnification	Does the software support zoom functions?
Convenient Performance	Does the software support drag-and-drop operations or point-and-click function on arguments elements with the mouse ?

User interfaces are generally designed to meet user requirements, this includes a capability to be attractive to the user. The *visual attractiveness* of the graphical view is another point evaluated here

by observing the use of the workspace, the assistance provided by the software to position the argument maps, to zoom on specific parts, to point-and-click with the mouse on argument elements and to drag-and-drop objects included into arguments.

Indeed, four parts of the user interface arousing interest were selected. These are **the menu**, **the toolbar icons**, **the panel for editing the diagram's layout** and the user's intended **feedback messages**. More than only evaluate alternately all parts of user interface cited above, the evaluator user must respond to a list of questions and assign for each a specific score within ten points. Table 6.11 gathers the specific questions list to assess.

Common and Additional Features

The salient features of argument mapping software are easily identifiable. Meeting the *suitability* criteria defined in ISO 9126 [21] as *the property that bears on the presence and appropriateness of a set of functions for specified tasks*, a list of the specific expected functions was established to serve as the basis of the assessment. Two particular aspects caught the attention : operations available on the processing of files as creating a new project, opening, saving or printing the argument result. Operations on the edition of argument components were also listed such as adding an element or a relationship, modifying its content and its properties, or deleting an element node or a link between two elements.

However, most software does not only implement these common features. Most of them provide very interesting additional features which facilitate the work of the user, such as exporting arguments screen shot into an html slideshow, or giving access to a tutor function where the professor can annotate or revise argument diagrams of his students.

Tool Proficiency

The tool proficiency is more than ever related with its simplicity combined with its learnability. Indeed, the *learnability* criterion is the *capability of the software product to enable the user to learn its application*[21]. This definition implies the learning curve at carrying out a task by a typical user as illustrated by Fig 5.12.

As with all new information technologies, the user will go through different steps of learning. The first one corresponds to the first contact with the software. At this point, the performance is very low on account of the discovery of available functions and operations. If user can benefit from a tutorial at the very first attempt he makes with the software, his learning level will increase quickly. A tutorial is a part of documentation strongly recommended for novice users. With many attempts, the learning growth is felt better, the user performances are more important. Finally, when the user regularly uses the software, he reaches a state called *"plateau"* where his performances will no more improve until some updates of the system occur. Then, the working of the tool can be changed. User should then learn the new features to use them correctly. This S-shaped curve of growth describes the difficulty with which a user learns to use the program. The steeper the curve is, the lower the difficulty level is because few trials are only necessary for better performance.

Unfortunately, the estimation of the specific learning curve for just one software is something very complex to establish. The purpose here is not to enter into the minds and to analyse the entire operation of learning. Our evaluation focuses mainly on the documentation made available for user in

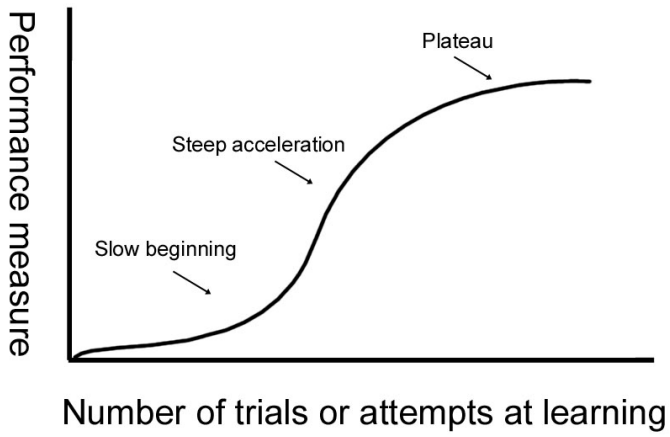


Figure 5.12: A typical illustration of the learning curve for new users

order to help him to achieve a minimal threshold of comfort in the tool proficiency.

Non-functional requirements or performance requirements enter also into account in this last category of evaluation about the tool proficiency. It specifies criteria that judge the operation of a system, rather than specific behaviours. The *reliability* and the software *maturity* are of the criteria that have not been studied so far. Reliability is the *capability of software to maintain its level of performance under stated conditions for a stated period of time.*[21] and its maturity is *its capability to avoid failure as a result of faults in the software*[21]. These criteria are studied by analysing errors met during the argument edition process and data loss caused by breakdowns.

The ultimate objective of this assessment is nevertheless to study the capability of the software product to enable the user to operate and control it. User must be able to perform in a rather straightforward way the task he has to done. The appropriate response has to be provided in processing time by the software product.

5.5.2 Arguments Design

After having assessed the argument software by three different perspectives relative to the user experience, the focus is on the argumentation assessment part. Each designed argument diagram is evaluated using two approaches. The first approach deals with the visual notations used and their cognitive effectiveness for understanding or achieving the argument diagram. The second approach discusses the choice for the underlying scheme of argumentation implemented before these notations. Both approaches increase the knowledge already acquired after the first part of the assessment on the user experience and improve specification data of an argument mapping tool that best satisfy the design of a security requirements argument.

Argument Notations

Visual diagrams are used to convey information. The representation of the information in a human-oriented way leads to an easier understanding of the information. During the reading, the perceptual processing of the diagram is the first phase followed by the cognitive processing. While perceptual processing is fast and automatic, cognitive processing is slower and sequential. In this section, criteria take into account the extraction of the information content from the visual notations composing diagram.

This evaluation of notations is done on the basis of a set of principles for designing effective visual notations defined in [32]. These principles, which are studied deeply in the scholar course entitled *INFO M431 - Ingénierie des exigences* are the *Semiotic Clarity*, the *Perceptual Discriminability*, the *Perceptual Immediacy*, the *Visual Expressiveness* and the *Graphic Parsimony*. In his article, Daniel L. Moody encompasses these measures in the concept of cognitive effectiveness defined as *the speed ease and accuracy with which information can be extracted from a representation*. These five principles have also been previously used in [31] to evaluate the UML visual notations.

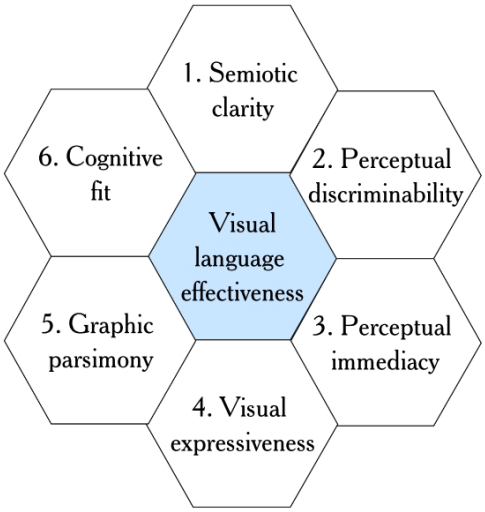


Figure 5.13: Principles for designing cognitively effective visual notations.

The visual language effectiveness is sometimes described with six main concepts as illustrated in Figure 5.13. The *Cognitive Fit* which wants that different visual dialects should be used for different tasks and/or audiences, is not relevant for the assessment because of the audience and the task have already been fixed in the assessment process. A clear description is given in the Tab 5.2 for the other five items.

The result of the evaluation of each of these criteria is a critical estimate. This can be a $++$ appreciation, which means that the principle is correctly respected in the notations achieved by the tool argument, or if not, the result would be the $--$ result. Between both, $+$ and $-$ are also available. The former means the property is relatively good, and the latter means that the argument notations do not support the principle enough.

Table 5.2: Description of visual effectiveness concepts.

Semiotic Clarity There should be a one-to-one correspondence between graphical symbols and their referent concepts in order to simplify use and minimise ambiguity. Anomalies can occur such as *Symbol Redundancy*, when multiple symbols are used to represent the same construct, or at the opposite a *Symbol overload*, when the same symbol is used to represent different constructs. Others anomalies such as the *Symbol Excess* or the *Symbol Deficit* which each has an impact on the complexity of diagrams, the former increases its complexity, while the latter reduces it.
Perceptual Discriminability Symbols used to represent different semantic constructs should be clearly distinguishable from each other. A correct interpretation of diagrams requires precise distinctions among symbols used in the scheme of the diagram. The *Visual Distance*, or the differences among visual variables, should be clear enough to avoid interpretation errors and ambiguity.
Perceptual Immediacy Notations should use graphical conventions that suggest the meaning of what they represent in order to enable to establish intuitively the meaning of a symbol. Providing cues to the symbol meaning, where it is possible, improves user intelligibility and his interpretation performance.
Visual Expressiveness Notations should use the full range and capacities of the visual variables. There are eight available visual variables: size, brightness, orientation, texture, shape, color, horizontal and vertical position, everyone with a different level of efficiency. For example, colour is one of the most cognitively effective visual variable.
Graphic Parsimony The number of different graphical conventions (visual vocabulary) should be cognitively manageable. Increasing the graphic complexity involves reducing the understanding of diagrams notations by users. The basic principle of parsimony is defined as: *Less is better*.

Argument Ontology

This last section focuses on the set of concepts belonging to the domain described by the underlying scheme of the tool analysed. The ontology *"refers to the content, to the objects and relations one uses to represent a domain, not so much to the symbols by which objects and relations are denoted"* [12]. After having evaluated argument notations in the previous section, the observation is here made on the argumentation theory used by the tool. The argument ontology defines many pieces of information such as the process for the problem elaboration, the proposals with supporting arguments, or the different kind of relationships between entities. The tool's behaviour is governed by rules or guidelines which lead to a correct construction of arguments implying the connection with the model supporting the tool.

This section turns therefore to know what are the patterns or the argument scheme underlying the argument mapping software. The effective concerns will be gathered in form of questions for each argument mapping tool, such as: Does the argument mapping tool support different types of argumentation scheme? Does the argument scheme prevent the program to perform any operations on argument elements? Does the model include additional interest such as weight preferences, or a value calculated by an evaluation process?

5.6 Summary

This chapter was the complete overview of the assessment process. We reviewed the necessary assumptions for completing the assessment. We defined the user profile, the task to complete, the information of each tool and the criteria. Now, lets enter in the part of the results of the assessment in the next chapter.

Chapter 6

The Results Description of the Assessment

After having deeply described the assessment process in Chapter 5, this chapter details the result of the assessment. Indeed, every tool mentioned in the previous chapter was concretly analysed by the user. This chapter is, therefore, the core of this assessment because it describes obtained results, firstly sorted by tool and divided in five parts following criteria such as defined in Chapter 5. The objective of this chapter is to share the observed characteristics of each tool in order to be able to discuss in the next chapter about what is good and what it would be better for such software tools.

The first part of this chapter resulted in a list of the observations for which a justified mark as objective as possible has been associated. This study would have been conducted on a group of people to avoid having problems with the subjectivity of the user, but to do that, it would have taken more time and resources. So this study is simply the opinion of one person who judges as objectively as possible all the criteria given with the knowledge this person has. The marks are referred to as benchmarks to measure the completeness of every approach on tools. At the end of the chapter, all these notations are gathered in a results review section organized in several tables to help the reader to identify the strengths and weaknesses of every tool.

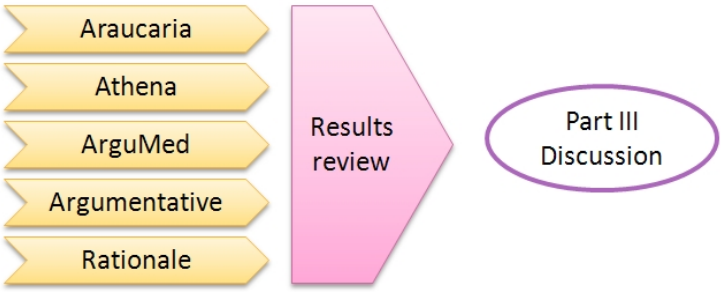


Figure 6.1: Order in which the tools are evaluated in this current chapter.

On Fig 6.1 is, therefore, illustrated the plan of this chapter by taking into account the order in which the tools are described and a preliminary comparison of the results leading us to the next part of the thesis : a comprehensive discussion about tools.

6.1 Araucaria

As presented in the previous chapter, the first software to assess is Araucaria, a very popular tool in the education field designed to build and analyse textual arguments structure. Security arguments are hardly taken into account by this tool because the model of argument, but before that discussion let us first start this review with the user interface study, its functionalities and the user experience.

6.1.1 User Interface

Broadly speaking, the user interface specifies how users will interact with the system. In the Araucaria user interface, the workspace's decomposition was identified in 6 different panels: the menu, the toolbar icons, the status toolbar, the graphical view, the textual view and the panel where appear the free premises before being added to the diagram. For the reader's understanding, a screenshot of the Araucaria main window is depicted in Fig 6.2. This figure shows a concrete representation of the tool treating a complete example of an argument diagram analysis.

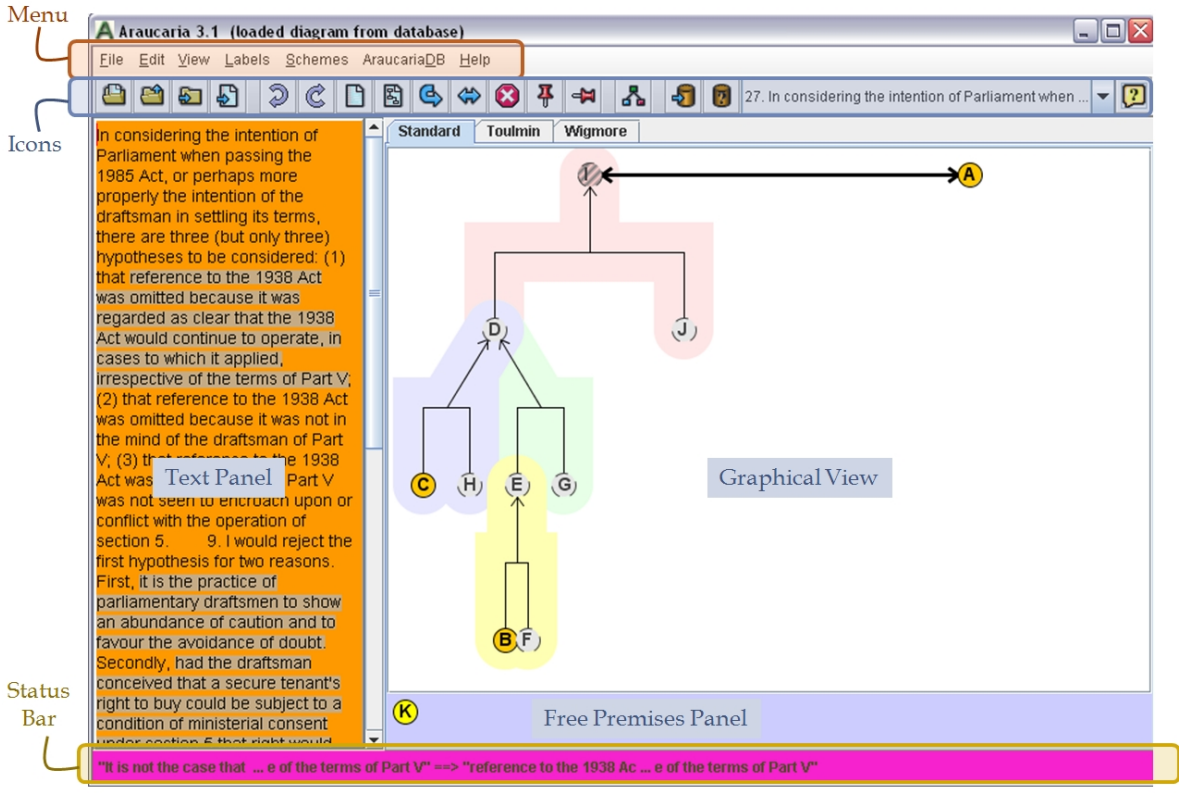


Figure 6.2: Araucaria 3.1 - Screenshot of the graphical user interface.

The different components of the user interface are shortly described in the next paragraphs.

Menu, Toolbar Icons and Status Toolbar

The main menu situated all above, connects the user to all features available in the tool. As illustrated on Fig 6.3, additional information included in tooltips related to the sub-menu elements appears when the mouse moves over. That enhances the easy-to-read and easy-to-understand information provided to the user. Furthermore, menu items associate a hotkey sequence with main features to make the software really convenient.

Figure 6.3: Araucaria 3.1 - Screenshot of the menu tooltips.

Just below the menu, the toolbar icons contains an icon for all the main functions available in the tool (See section 6.1.2 for available functions). The user could appreciate this completeness. The status toolbar or information toolbar, at the very bottom of the main window, displays premise content of selected item and other extra information such as the reason when an operation failed, the confirmation of a file saving, etc.

Menu : 9/10	**Toolbar Icons : 9/10**	**Status Bar : 8/10**

Textual and Graphical View

The main part of the workspace is devoted to the textual and graphical view of the argument. The linking between the text part and the graphical part implies almost inevitably the building of the argument on the basis of a text file. The textual argument description cannot be edited with the Araucaria software. On the graphical view, the right click on a selected argument component brings up a new menu with appropriate functions. Each argument component is automatically positioned on the view and no drag-and-drop function is implemented to allow users to rearrange boxes on the chart. Tabs above the graphical view are argument styles; they concern the argument ontology and will be discussed in the section 6.1.5.

The view menu allows user to alter how much of the diagram can fit in the window using different views. The *full text* view, adopted for its clarity, displays the full text of every node. The *full size* view displays only nodes' ID to save space as illustrated in Fig 6.2. Finally, the *scaled* view fits the whole argument analysis in the window. This last option helps to display large size arguments and to manage their complexity.

Graphical View : 5/10

Other

Predefined questions on user interface convenience (See Appendix A.1) reached an average score of 6.5/10 to which 2 points are added for the customization function and the easy database access. In fact, the possibility to change colour background settings is another element in the Araucaria implementation which seems to be appealing for some users but useless for others. Setting the user's preferred colour in the background of each panel of the interface is a first feature in the software personalization process designed for the comfort of the user. The easy way which the user can select a database

element is another advantage of this implementation, indeed a list with the user research results is included in the toolbar icons.

User Interface Convenience : 6.5/10	Other : +2

Let's turn now toward Araucaria features.

6.1.2 Features and Computational Services

In this section, we observe two different angles of the available functions of the Araucaria tool. Firstly, we focus on expected functions and after that we look closer to additionnal features provided by the system.

Expected functions for argument mapping tools.

Although a function to create a new project or a new argument is missing, it is still possible to build a new argument map without opening any text file simply by adding missing premises, called an *enthymeme*. The objective of this tool is not to create an argument map from scratch but the user can appreciate the feasibility of the operation. Another point appealing attention is the AML format. Argument Mapping Language is the special format in which argument files are saved, it was created specially to answer the need of Araucaria to build argument files and is defined in a DTD file available for other developers. Moreover, the path of recently opened files is stored in memory to be reopened more easily and quickly. The number of considered recent files is configurable.

However, the absence of the printing feature is regrettable. Saving the argument map as an image and then print it, seems to be the nearest feature to get the same result as a printing.

See Table 6.1 for a clear summary of expected features concerning argument files.

Table 6.1: Araucaria 3.1 - Description of the file processing features.

Files Features :	
▷ Creation	From a text file and also from scratch
▷ Opening	· A text file, · An argument file formatted in Argument Markup Language (AML) based on XML, · An argument from Araucaria online database. · Consideration of recent files.
▷ Saving	· As an image (tiff, jpg), · In the AML format, · In the Araucaria online database.
▷ Printing	/

Concerning argument operations, the software implements all of required functions as adding a missing premise, deleting selected items, undo and redo features, linking and unlinking selected statements, editing content statement, erasing argument diagram, etc. Premises can also be toggled into a refutation.

Expected functions : 8/10

Lets see what the tool offers as less expected features.

Additional features

An access to the Araucaria database is available for all users. It is another mean to seek arguments achieved by other users and to share his argument diagrams. The program also includes a tutor mode to be able to compare two different diagrams. The software system computes then the percentage of matching between premises and vertex of both files. Obviously, the ability to add, open and save user own scheme sets is an essential additional feature to this software. Finally, in the properties item of the main menu, statistics detailing the current task can be found such as the number of words in the text, the number of enthymemes, propositions and refutations included in the argument, etc. The file's properties, such as the author, the date, the text's source or comments can be edited.

Effective additional features : 7/10

6.1.3 Tool Proficiency

Help file, manual document, tutorial and updates

Araucaria developers provided a manual document and a help file under the form of html files stored in the software folder, directly accessible from the tool. The manual is downloadable on the website[1]. The sixty pages of the manual are really complete, describing all functionalities available in the tool. The help file on the other side, gives only more explanations about menu items. A tutorial has not been written yet. No specific feature is implemented to keep the software updated but user might visit the website for news and more information. The website is also the perfect place to find other resources as online argument sources, argumentation scheme sets, the AML DTD and AML samples.

Documentation : 8/10

Errors and difficulties

During the use of Araucaria to achieve the authentication diagram, text display troubles were discovered in the graphical view panel as refresh lag of textual content in graphics. That causes a lot of frustrations and confusion to the user leading to a negative user experience. Saving diagrams in JPG picture encounters also problems. Besides this, it was noticed that unlinking a premise from the one it supports is not feasible, therefore, if the user makes a mistake when linking two premises, he has no other choice than to delete the wrong one and to create another free premise with the same content rather than to be able to alter the link.

Overall, the weaknesses described above reduce moderately the level of user experience provided by the Araucaria software.

Reliability rate : 6/10

By this comprehensive analysis of the tool experience provided by Araucaria, we considered the context implemented to complete the task of a security argument. Now, lets examine the argumentation conventions followed by the Araucaria tool based on the carried work.

[1] `http://araucaria.computing.dundee.ac.uk/`

6.1.4 Argument Notation

After the design of the argument task with the Araucaria system, the resulting diagram is given on the Fig 6.4.

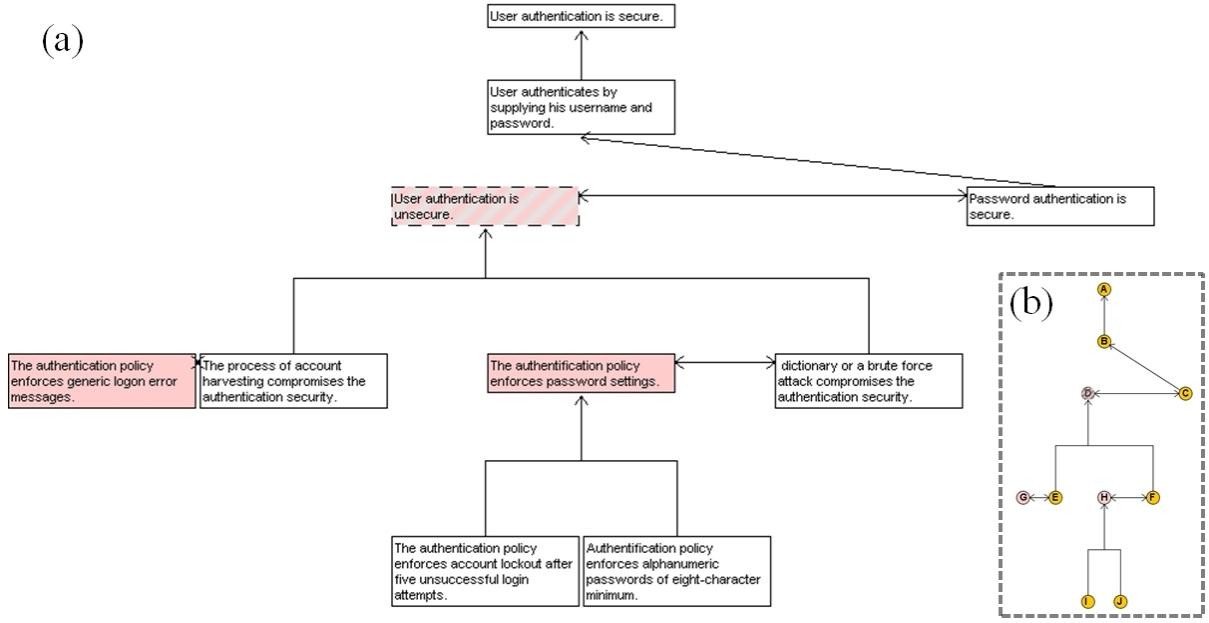


Figure 6.4: Araucaria 3.1 - Representation of the argument task.

Figure 6.4.(a) illustrates the argument diagram obtained with the Araucaria tool according to the *Full text* view by the zoom function. Figure 6.4.(b) shows another view of the argument, the *Full size* view, more concise to present the global aspect of the argument diagram.

The set of notations used by the standard style of Araucaria is illustrated in Fig 6.5. Premises from the text are in white boxes connected by a simple arrow to statement it supports. Refutations statements are in red boxes connected with a double-headed arrow to only one premise. The grey boxes with dashed borders indicate a missing premise, not directly created from the text but added by the user. And finally, missing premises toggled in refutations are also bordered with dotted lines with a background texture which consists of red and grey oblique lines.

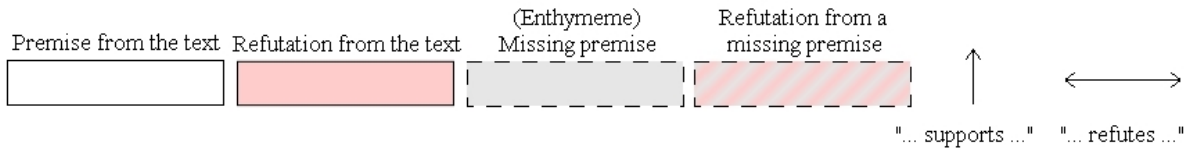


Figure 6.5: Araucaria 3.1 - Visual components of the system.

Let's study now the cognitive effectiveness of the set of notations using the five criteria described in Section 5.5.2. Table 6.2 gathers an appreciation rate and the explicative information for every

criterion.

Table 6.2: Araucaria 3.1 - Assessment of the visual language effectiveness.

Semiotic Clarity	−	Observation of the diagram notations brings up the lack of a specific graphical symbol for expressing the conclusion concept which is only expressed by a premise supporting another premise. This is a symbol deficit for the conclusion concept, and an overload of the premise symbol.
Perceptual Discriminability	+	A colour differentiation of the background has been chosen in order to help the user to directly filter premises and refutations.
Perceptual Immediacy	++	Even if notations used to represent argument are simple boxes and arrows and the rectangle shape is one of the most common form to express any idea, the perceptual immediacy is in fact very good because of the use of vertical position to express that one premise supports the other or the conclusion, the horizontal position, the red colour and the double-headed arrow to mean the refutation between two premises.
Visual Expressiveness	++	Various visual variables were used to distinguish diagram components, a total of four : the vertical and horizontal position, the texture and the colour.
Graphic Parsimony	++	Number of graphical conventions is low because there is only four notations available for the representation of graphical nodes and only two possibilities for relationships.

The cognitive effectiveness is relatively sufficient mainly because of the vertical positioning and the red colour used in the diagram representation. Let's now look closer at the last assessment's point : the characterization of the argument scheme.

6.1.5 Argument Ontology

The standard style predefined in the Araucaria software is based on a representation of argumentation by using diagrams consisted of numbered statements and arrows indicating support relationships. This style is known as the Beardley/Freeman model in [18]. Other styles, not used during the exercise, but available in the tool are the Toulmin structure and the Wigmore model (See Sections 2.5.1 and 2.4.2 for more information).

Furthermore, the tool allows creating user own argument schemeset. Other argument schemesets met through argument maps discovered in the database have specific types as abductive argument scheme, from cause to effect argument, from expert opinion, popularity, example, analogy, etc.

More than the content statement of each premise or refutation node, additional textual information can be added to appear in the diagram. The ownership information assigned to the proposition can be displayed in the node under the premise content in a green background. That gives the ability to

represent multi-party arguments. Evaluation information can also be noted just above the ownership tag in a pink background as illustrated in Fig 6.6.

Figure 6.6: Araucaria 3.1 - Illustration of ownership and evaluation information.

In order to express our security argument task, we encountered some difficulties. Indeed, Araucaria does not currently support a single premise supporting multiple conclusions (known as divergent argumentation). In other words, no one node, neither the premise nor the refutation node, may support more than one other node. Moreover, a premise can only be linked with at most one refutation. In our case, two different refutations encountered the premise *Password authentication is secure*, so a missing premise was created and toggled in a refutation to be the main refutation supported by both others.

Both premises are not reported as refutations too but only simple premises supporting the main refutation. Therefore, after rebutting them with security requirements, these requirements were then denoted as refutations, which is somewhat which seems weird for the reader. Furthermore, the refutation of a refutation is not taken into account by the model of this tool. Finally, the refutation relationship seems to be reciprocal between both propositions because of the double-headed arrow, but it is not the case. The red node rebuts the other and not the opposite.

For summarizing what we discovered during the assessment of the Araucaria 3.2, we can say that despite a very rudimentary level of the user interface, the software tool fulfilled its function of creating a simple argument without rebuttal of rebuttal component and on basis of a text. This concludes the analysis of the Araucaria system. Let's look closer at the next argument mapping tool : Athena Standard.

6.2 Athena Standard

Athena is overall used for reconstruction of philosophical arguments, operated on laptop computer connected to a projector in front of a class of students. Establishing security requirements seems so far from teaching and learning argumentation theory as Athena software is designed for. However, the ease with which Athena enables argument conception, made of this software application an excellent tool to assess.

6.2.1 User Interface

Athena Standard is another tool among software packages available on the market that facilitate argument diagramming and learning of argumentation theory. Araucaria software, already evaluated, is also this kind of software. For a student, understanding a complex argument found in philosophical texts, is a task made easier thanks to a diagram. Athena offers several different procedures controlled by the user from the interface.

The user interface of Athena is illustrated on Fig 6.7. As for each tool, let's go over the different panels of the interface.

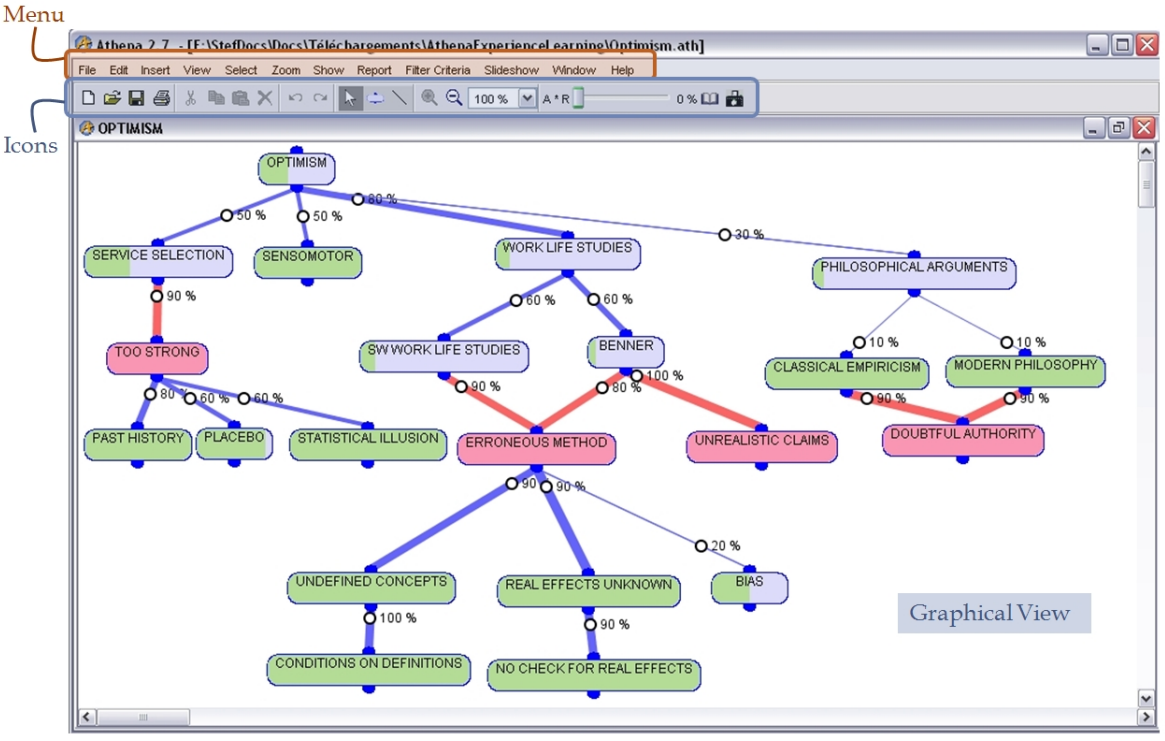


Figure 6.7: Athena Standard - Screenshot of the graphical user interface.

Menu and Toolbar Icons

Absence of a status bar at the bottom of the software is the first detail to be noticed and is a disappointing aspect. Besides this, the menu is expanded and well-supplied, giving access by categories to all operations available with the software. Hotkeys are used and their sequence of buttons is presented on menu items. Concerning the toolbar icons, icons are also integrated in sub menu list display. Furthermore, a mouse-over on an icon triggers display of a textual message in a tooltip corresponding to the operation realized by the icon. Operation not permitted in the current state of the task are represented by unavailable icons. The zoom option is directly available from the toolbar icons in addition to the rate of the filter criteria. The mouse cursor can be changed in a node element or a connexion serving to be positioned directly inside the diagram window with the mouse.

Menu : 8/10	**Toolbar Icons : 8/10**

Graphical View

Athena graphical view is an edition window where user can drag and drop all components of the argument diagram with the mouse. Indeed, the software is not automatically assisted to enhance the rendering of the display of argument diagrams. That permits a greater degree of freedom in the arrangement of argument nodes on the screen.

Double-clicking on an element leads to the opening of a window describing the node properties where user might encode the node name, comments specification and its acceptability level. Point-and-click and drag-and-drop features are extremely useful and user-convenient to create, select and move diagram elements. Athena application provides also a zoom feature. The zoom magnification is expressed as percentage of native resolution on a scale from 10% to 100%.

Graphical View : 8/10

Other

Predefined questions on user interface convenience reached an average score of 6.5/10 (See Appendix A.2).
To this result 1 point is added for a specific feature which manage the display of projects. Indeed, the graphical panel can be divided in several windows including one open project. Athena software provides a function for arranging all these different windows in several different ways, such as Cascade, Tile All, Tile Horizontal, or Tile Vertical.

User Interface Convenience : 6.5/10	**Other : +1**

Let's turn now toward the specific features available in Athena.

6.2.2 Features and Computational Services

Here is an overview of expected and additional feature the user could find in the Athena Standard software.

Expected functions for argument mapping tools.

A part of developed features in Athena Standard are basic such as the feature to create a new athena project , open it , close all, save it as an athena file or an athena template, print it or print the argument tree or sub-tree includes in it. However, the Athena software does not take into account other argument formats of file.

See Table 6.3 for a clear summary of expected features concerning argument files.

Table 6.3: Athena Standard - Description of the file processing features.

Files Features :	
▷ Creation	Of a new project from scratch by creating nodes and connections between them.
▷ Opening	· An argument project under the athena format (*.ath). · A template in a athena format file
▷ Saving	· As athena argument file. · As a template.
▷ Printing	A whole project or a tree or a sub-tree can be sent directly to the printer.

Main features of diagram edition are implemented such as inserting a node or a connection, cutting, copying, pasting or deleting components, selecting all components and also undo and redo functions. A zoom function is also available by decreasing or increasing zoom percentage to adapt the view. Many projects can be opened in the same time and the tool provides some buttons to arrange the different windows.

Expected functions : 8/10

Additional features

Three specific additional features must be explained to be appreciated.

The first one is the display filter applied on argument nodes following a certain threshold of relevance and acceptability given by the user. The weakest arguments are filtered out by the feature if acceptability or relevance or their product is less than the given threshold.

The second feature is the generation of an html report of the current or final state of the project. Report assistant allows the user to customize the report by giving information, for example the headline, the author(s) name and mail to contact, the background of the argument diagram, but also by configuring the type of the report, the orientation of the page, elements to include inside, the outcome of structure check, of acceptability and relevance check and robustness check. All that information is included with current diagram information in the generated report. User can also find the implementation of a teacher report feature with positive and negative comments.

Finally the third feature is the automatic generation of a slideshow from many screenshots of whole argument or active window taken during the argument construction. A particular name is asked for every screenshot, so during the generation, user can select the different PNG files he want to include in the slideshow. He can also add text files to display in the slideshow too. The result of the generation is a repertory containing all necessary to display the slideshow in an index.html file.

Effective additional features : 8/10

6.2.3 Tool Proficiency

Help file, manual document, tutorial and updates

The menu item about help provides to user the access to a tutorial of the simple operations such as how to create a new project, an argument tree, nodes and connections, how to change nodes and connections properties, but also how to use additional features explained above, the filter functions, the report function and the slideshow function. Another item in the help menu is a direct link to the online website of Athena[2]. Many scientific papers are available on this website, including several written in Swedish, but no manual has been found.

Documentation : 6/10

Errors and difficulties

Some displaying troubles were discovered during the utilization of the software but nothing that completely prevents the diagram building. The properties of a connection do not appear on the screen. The printing is sometimes problematic in the sense that it lacks a little piece of the diagram at the bottom of the sheet. However, we can consider this program as very reliable.

Reliability rate : 8/10

The Athena tool experience was analysed, lets now look closer at the argument conventions implemented by the system.

6.2.4 Argument Notation

Fig 6.8 illustrates the argument diagram achieved with the Athena Standard system. The Athena system provides to the user an evaluation process, where he can assign an acceptability rate to each premise and a reliability rate to connections between premises. The security requirements task was designed without working with the evaluation process, it is the reason for what this diagram is almost colourless. The evaluation process gives more impact to the diagram visualisation because it colours nodes and links in green or red. But it was not necessary for the current task.

[2]http://www.athenasoft.org

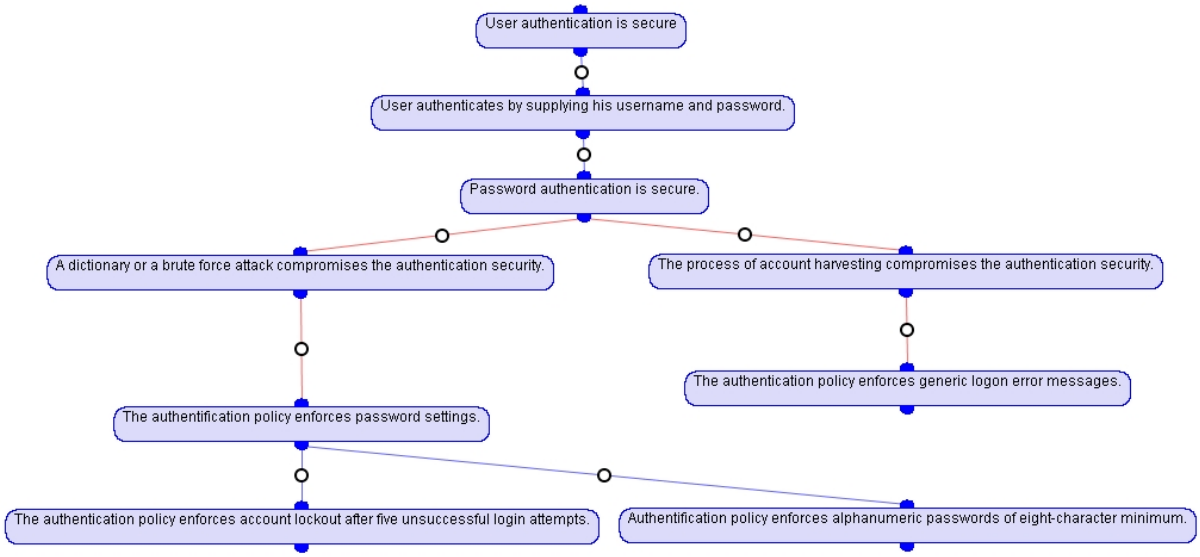


Figure 6.8: Athena Standard - Representation of the argument task.

The set of notations used for forming the diagram is described on Fig 6.9. A differentiation has been done between notations used with or without the evaluation process. On the picture, it is easy to notice that the node without neither green nor red colour background has several underlying concepts. In fact, its meaning is understood following its position in the argument diagram.

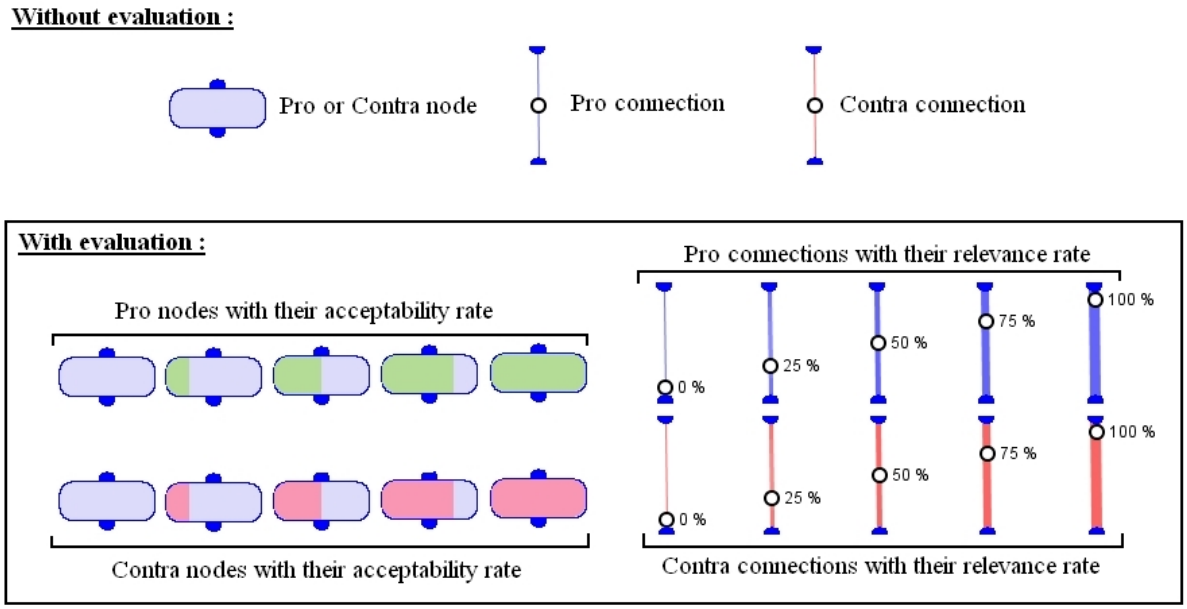


Figure 6.9: Athena Standard - Visual components of the system.

Without the evaluation process, if the node is connected with a contra connection at the top point, it means this node is a refutation, in the other case, it is a supporting premise. Indeed, each node has

two connexion points, at the bottom of the node symbol and at the top. It is impossible to connect two nodes from and to the same connection point. The connexion point where the node is connected with others is important during the evaluation process. Even if there is no arrow on the connection link, the support or the rebut relationship is going from the bottom to the top.

The system was designed for a first step of argument creation and a second step of argument assessment. The creation of an argument diagram without any evaluation brought to the user some simple, or maybe too simple notations conventions. Differences of argument notations before and after the evaluation process is the main proof that the system is mainly oriented on the evaluation process. With the evaluation process, argument notations are undoubtedly more intuitive, more complete and more colourful for the understanding of the user.

Let's continue with Table 6.4 to clarify some design defaults or Athena Standards which represent obstacles against the cognitive effectiveness of the reader.

Table 6.4: Athena Standard - Assessment of the visual language effectiveness.

Semiotic Clarity	$-$	As it is possible to see on Fig 6.9, a symbol overload occurs with the simple node symbol. This does not make the reading of the argument diagram easy.
Perceptual Discriminability	$+$	Although all nodes are similar before the evaluation process, the perceptual discriminability is better during the evaluation process. Indeed, at the evaluation process, a colour differentiation of the background and a certain rate of colouration related to the acceptability rate have been chosen in order to help the user to directly filter the nature and the importance of premises.
Perceptual Immediacy	$-$	Only two visual aspects help the user intuitiveness to read the argument diagram : the red colouration of the refutation connection and the particularity of both points of connection on the premise node. Otherwise, connections are not oriented, nodes are not positioned in a way that user can understand intuitively the relationship between two premises.
Visual Expressiveness	$--$	In the diagram version without evaluation denotations, only the colour of the refuting connection helps distinguish some diagrams concepts. With evaluation colour, two visual variables are present, the colour of nodes to express their nature and the line breadth to display the strength of relevance. A textual information is added to connections. Only two visual variables is judged as insufficient.
Graphic Parsimony	$+$	Before evaluation, the graphic parsimony is correct, but the graphic complexity increases in the evaluation process by taking into account acceptability and reliability rates in the graphical conventions.

The cognitive effectiveness is pretty bad because of the lack of visual differences between symbols and the use of only two visual variables which is null when the evaluation notations are not used. Let's now look closer at the last assessment's point : the argument ontology.

6.2.5 Argument Ontology

Premises relationships are broadly allowed as a thesis can be supported by premises that, in their turn, can be supported by premises, a premise can support or refute more than one other premise and a premise can be refuted by several premises.

The evaluation feature is calculated according the acceptability rate of premises and the relevance rate of connections which are both essential concepts in this ontology. A complete evaluation results in a tree graph with all values assigned. The acceptability rate is a measure of the user's force to hold the conviction represented by the box. Therefore, the argument objective is to show how the acceptability of the premises make the conclusion also acceptable. The relevance rate indicates how much support a superior conclusion maximally can draw from a subordinate premise. Unfortunately, the specification of the security argument to complete does not include this information so the task was rather limited.

A summary of this tool assessment could be that the tool provides what is necessary for the construction of the argument despite wrong notation conventions of the argument components. Let's look closer at the next argument mapping tool : ArguMed 3.

6.3 ArguMed

The ArguMed-system is the third software to be evaluated. This tool was mainly designed to meet the needs of lawyers and other persons who are trained to argue in cases of law. At first glance, it appears smaller in terms of functionalities than the previous tool. Let's see exactly what it is about.

6.3.1 User Interface

The assisted organization of the workspace reserved for the argument map seems to be more efficient to display complex arguments. These elements are visible for the reader on the screenshot in Fig 6.10. The example treated here is one among many delivered in the archive package of the software to help users to take complete control of the ArguMed software.

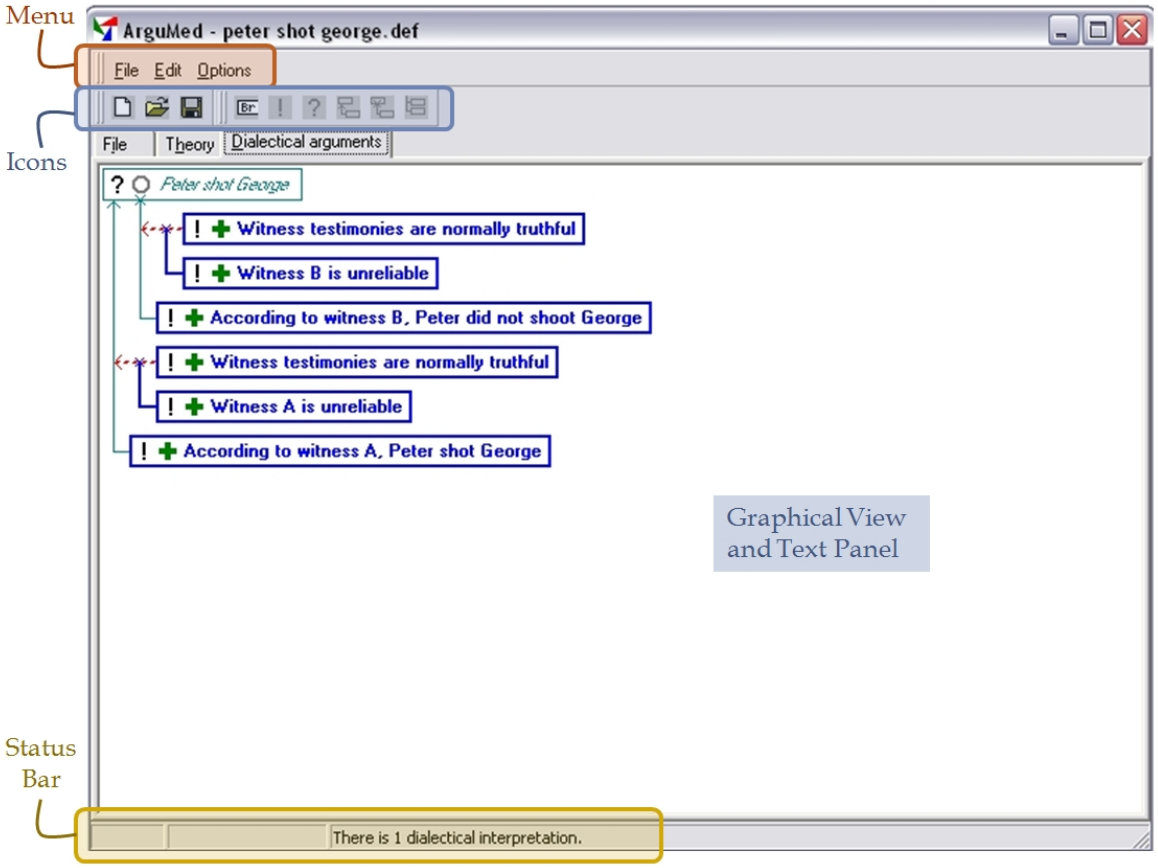


Figure 6.10: ArguMed 3 - Screenshot of the graphical user interface.

The objective of this first part of the assessment is to formulate as much as possible interface properties.

Menu, Toolbar Icons and Status Toolbar

The menu is subdivided into only 3 items, the tool does not dispose of additional information to advise the user on the utilization of features. The trial-and-error technique appears to be the best one in the first use of the tool. The menu is very simple, unsophisticated, for example hotkeys are not implemented. The menu offered just what it has to without any improvement.

The toolbar icons resumes most but not all features carried out by the tool. Concerning the choice of icons, the icons used in this toolbar remind icons used in old-fashioned Microsoft Windows tools. As already seen in Araucaria system, a small textual field appears when mouse moves over every icon in order to describe textually the function hidden behind these.

Figure 6.11: ArguMed 3 - Screenshot of the status toolbar for a selected component.

Figure 6.11 illustrates detail given in the status toolbar when a box is selected. The first field shows a pair of numbers indicating the position x and y in the point-and-click diagram panel of the interface. The second field gives the type and the status of the selected statement, and the last field displays the text content.

Menu : 5/10	**Toolbar Icons : 7/10**	**Status Bar : 7/10**

Graphical View

ArguMed uses a mouse-sensitive screen. Double-clicking on a simple statement allows its edition in an easy and fast way. The simple point-and-click feature on an element gives information in the status toolbar and the right-click displays the same result than the *Edit* submenu for the element. The system determines how the argumentative data are organized on the screen. This automatic tree disposition of argument components is very clear and allow user to work with complex and large scale arguments while keeping always the control. There is no graphical argument view strictly speaking, only a tree view of the argument.

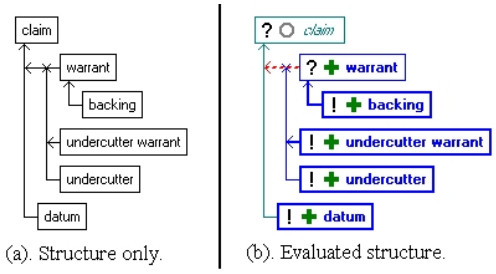


Figure 6.12: ArguMed 3 - Both styles of argument with or without evaluation.

Furthermore, as illustrated on Fig 6.12, an option is available to display the argument with or without the structure evaluation information.

Graphical View : 7/10

Other

The average score to predefined questions on user interface convenience is about 6/10 (See Appendix A.3).

User Interface Convenience : 6/10

Let's turn now toward ArguMed functionalities assessment.

6.3.2 Features and Computational Services

The ArguMed-system offers only the most basic features to the user. About files operations, it includes the creation of a new argument tree, opening and saving the file under the deflog format (*.DEF). Concerning operations about argument components, the software allows adding, editing, deleting elementary sentences, altering its status and its type and joining it a new support or attack sentence. Particularly, adding relationships between statements is based on templates (support or attack).

Table 6.5 resumes expected features concerning argument files.

Table 6.5: ArguMed 3 - Description of the file processing features.

Files Features :	
▷ Creation	· Adding an elementary sentence. · Based on templates (support or attack).
▷ Opening	Argument formatted into deflog file (*.def).
▷ Saving	Argument formatted into deflog file (*.def).
▷ Printing	/

This version of ArguMed system is evaluative : the status of statements and arguments can be determined by the system. ArguMed uses the logical semantics of DEFLOG for the evaluation of the arguments[3].
A dialectical argument can have any number of evaluations, in fact there can be no evaluation, or one, or several. Evaluation of dialectical argument occurs automatically in the background. Evaluation process is calculated again whenever changes occur in the diagram.

Expected functions : 7/10	**Effective additional features : 7/10**

[3] See [57] for more information on the logical system DEFLOG

6.3.3 Tool Proficiency

Help file, manual document, tutorial and updates

User might only read scientific papers written by the tool developer, Bart Verheij, such as [57, 56] as tool support and documentation. This reading does not probably aim any kind of audience but rather particularly researcher and university staff. These documents explain primarily the argumentation theory behind the tool, the evaluation rules, the program design, etc. and give also many points of comparison with previous versions of the tool.

Documentation : 7/10

Errors and difficulties

No critical error has been met during the use of this program, features are simple and efficient.

Reliability rate : 10/10

6.3.4 Argument Notation

ArguMed does not offer the alternative to create a diagram but only an argument tree, so the tree related to the security authentication task is illustrated on Fig 6.13. The first thing the reader can notice is that a support or a refutation can cover both the node or the connection between nodes, as it is the case for the premise *Password authentication is secure* which supports the connection between *User authenticates* and *User authentication is secure*.

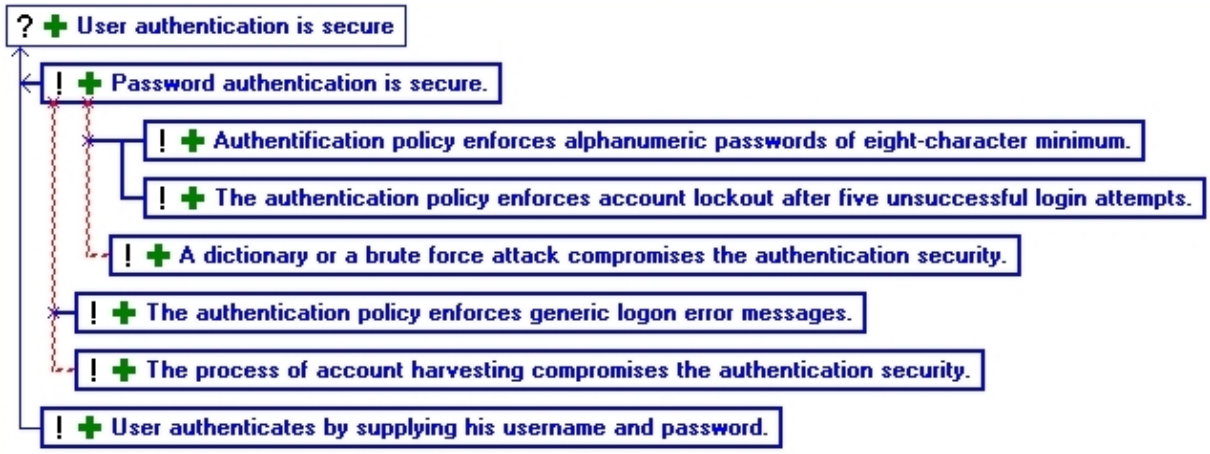


Figure 6.13: ArguMed 3 - Representation of the argument task.

Fig 6.14 gathers symbol notations and their underlying concepts. For premises, five symbols are combined to express the six possible results. Exclamation point means the assumption type while the question mark is the sign of an issue. Issues and assumptions have a given status. The status is either *justified* with a red plus symbol, either *defeated* represented by a red cross or either *open* with a grey circle.

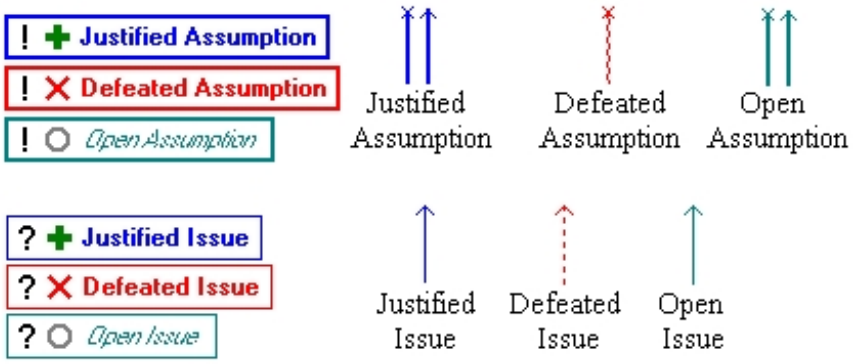


Figure 6.14: ArguMed 3 - Visual components of the system.

Let's study now the cognitive effectiveness of the set of notations using the five criteria described in Section 5.5.2. Table 6.6 gathers an appreciation rate and the explicative information for every criterion.

Table 6.6: ArguMed 3 - Assessment of the visual language effectiveness.

Semiotic Clarity	+	The one-to-one correspondence is well respected for nodes characteristics. For connections links, some concepts can have two different ends of arrow according to the attack or the support function toward the conclusion.
Perceptual Discriminability	++	Visual distance between exclamation point and question mark is good and this between the green plus, the red cross and the grey circle is very good. Symbols are very clear and easily differentiable.
Perceptual Immediacy	++	Red cross in a red rectangle suggests intuitively the defeat concept to the user mind and the green plus symbol is also quickly understood. The head of the arrow is different depending on the relationship it represents.
Visual Expressiveness	++	The main visual variable used is obviously the colour although the vertical and horizontal position are also used to improve the level of efficiency of the user understanding. The texture of the arrows is also a significant variable used in these visual conventions.
Graphic Parsimony	++	The complexity of the visual vocabulary is manageable for an inexperienced user.

The cognitive efficiency of the choices of notation conventions implemented by the developers of ArguMed, is very good and facilitates a quick understanding of the argument for the reader.

6.3.5 Argument Ontology

Argumentation scheme implemented here to formalize the theory is obviously the form of defeasible or dialectical argumentation. As we have seen in the argument notations analysis, the type of each statement (issue or assumption) is indicated by a symbol before each statement with the justification status (justified, defeated or neither). Some rules define the statement status. A statement is justified if and only if either it is an assumption, against which there is no defeating reason, or either it is an issue, for which there is a justifying reason. A statement is defeated if and only if there is a defeating reason against it. A reason is justified if and only if the reason and the conditional underlying the corresponding supporting argument step are justified. A reason is defeated if and only if the reason and the conditional underlying the corresponding attacking argument step are justified.

The system permits both forward and backward argumentation or, in other words, inference and justification without having to establish a central issue. Therefore, it is possible to draw conclusions from premises, to adduce reasons supporting issues, and also to turn the mentioned conclusions into a reason for another premise.

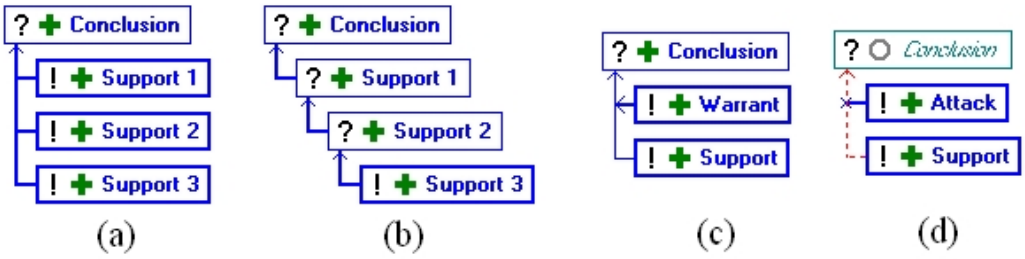


Figure 6.15: ArguMed 3 - Four different visual basic relationships.

As illustrated on Fig 6.15, a conclusion can be support in two different way, either by coordination between several premises in parallel as shown on the (a) part of the picture or either by subordination of a chain of supporting premises such as on the (b) part of the picture. The (c) part of the picture illustrates the notion of the warrant underlying argument steps, while the (d) part describes an attack of the conditionals underlying supporting argument steps.

To summarize this assessment, the possibilities that are offered by the ArguMed 3 software on the point of view of the user experience are limited, while the argument tree is designed with excellent visual conventions which significantly improve th e argument reading.

6.4 Argumentative

The penultimate tool to be assessed is Argumentative, an Open Source software product to put together an argument visually. This software seems really effective from the first use with an argument tree and a diagram argument.

6.4.1 User Interface

As you will see soon in the continuation of your reading, the Argumentative system has a very powerful interface. Let's first look closer at the screenshot (Fig 6.16) before to comment each panel as we have already done for the previous tools.

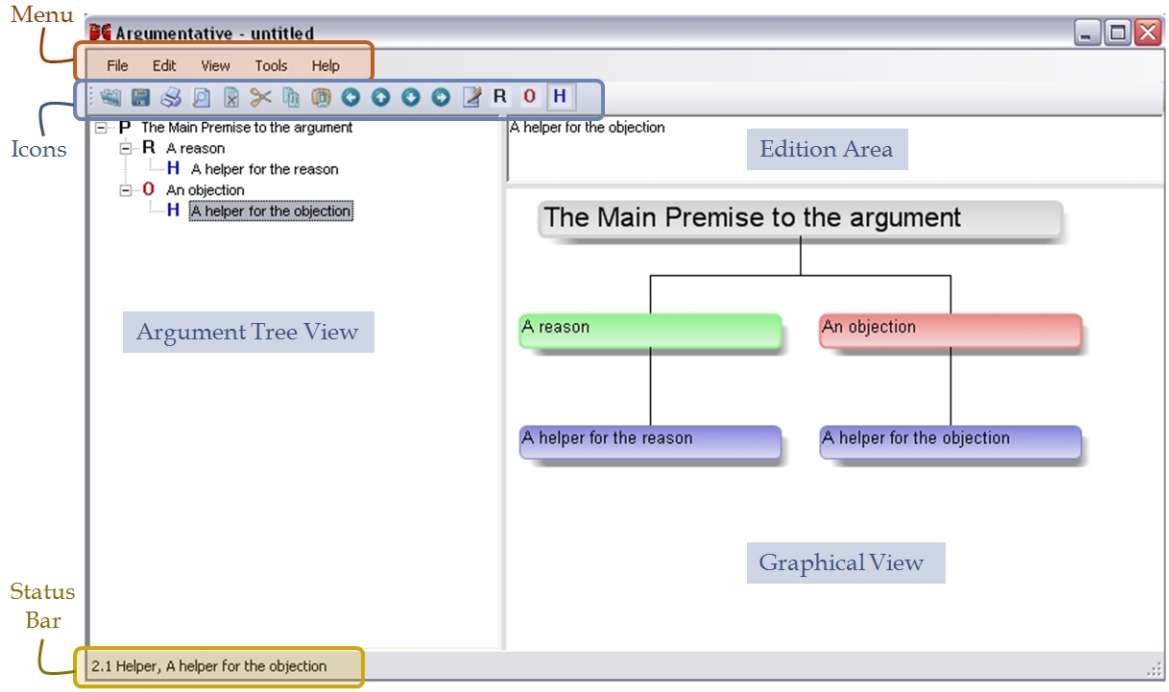


Figure 6.16: Argumentative - Screenshot of the graphical user interface.

Menu, Toolbar Icons and Status Toolbar

The main menu of Argumentative software is well-supplied with all functions implemented by the tool. Using hotkeys sequences in submenu items gives to the user another way to enhance the user experience. Additional information is provided for only a few of functions such as *Export...*. The *View* menu item allows to disable the display of the graphical view, the edit area and the toolbar icons. Toolbar icons allows quick access to common functions, displaying a textual field of information when mouse moves over each icon. Icons pictures are fresh and appealing for the user, and include all the main functions of the software. The status bar displays relevant information on a selected statement such as its textual content and its position in the tree, and gives also additional information on the zoom scale or the graphical view selected.

Menu : 8/10	Toolbar Icons : 9/10	Status Bar : 8/10

Tree, Edition Area and Graphical View

The tree view is similar to a directory view in windows file explorer. Arrow keys allows to move in the tree and to expand or contract subarguments. This last operations are also available by clicking on the $+$ or $-$ with the mouse. Edition area allows the manipulation of textual content statement. The graphical view is parametrizable under the Options item of the Tools submenu. The point-and-click feature on an argument component implies its selection in the argument tree view. Limits between these tree panels can be modified to enlarge a panel or reduce another.

Argumentative software provides many options in the *View* menu item to allow the user to choose the display of the argument in the graphical view. Among them, the zoom function allows limited magnification options for the graphical view from 25 to no limit of % with a full view for fitting the map in the graphical view. Another submenu item is the *Quick View* which contains some predefined options of the argument display in the graphical view. The *Orientation* item defines the direction in which the diagram is positioned.

Graphical View : 9/10

Other

Predefined questions on user interface convenience (See Appendix A.4) reached an average score of 9/10 to which are added 2 points for the optional parameters available in the *Tools* menu item to customize completely argument maps. For example, it is possible to alter the layout of boxes, their line width, their box colour or the set font. It is also possible to change the joining line style between arguments in the graphical view and others parameters like that.

User Interface Convenience : 9/10	**Other : +2**

Let's turn now toward Argumentative available features.

6.4.2 Features and Computational Services

The Argumentative software was designed to be compatible with multiple formats of argument in addition to its own, the AXL format. So it enables to open argument files created by the Araucaria software, another software already evaluated; Reason!Able, the ancestor or Rationale; and finally Argumentative is also compatible with Rationale files.

The diagram saving feature is largely implemented by including possibility for the user to save his argument as an image in multiple formats such as JPEG, PNG, GIF or Bitmap, and a menu item is available to make the opening of recent files easier.

An additional feature, not found inside other tools, is the export feature to a MS Word or PowerPoint file without need of any settings.

See Table 6.7 for a clear summary of expected features concerning argument files.

Table 6.7: Argumentative - Description of the file processing features.

Files Features :	
▷ Creation	· Based on a basic diagram to edit and complete. · Enter to create same type as current component.
▷ Opening	· Argumentative files (*.axl). · Rationale files (*.rtnl). · Reason!Able files (*.re3). · Araucaria files (*.aml) and scheme sets (*.scm). · Recent Files.
▷ Saving	· Argumentative files (*.axl). · Rationale files (*.rtnl). · As picture (Bitmap, PNG, JPEG, GIF).
▷ Export	MS Word, MS PowerPoint.
▷ Printing	Any files with Preview and Page Setup.

Operations related to argument components are similar to those met in others tools, such as properties consultation, adding, editing, deleting an element, cut/copy/paste and the undo feature are all available. Among additional features, the availability of a find/replace function and a spell-checker is to be noticed.

Furthermore, Argumentative's developers have also implemented a transform feature which takes a structured argument and transforms it into other formats using an XSLT stylesheet.

The amount of options settings is prominent and plays an essential role in the possibility for the user to customize diagrams appearance. User preferences might even be stored in a XML file to be re-established at the next software start-up.

Expected functions : 9/10	**Effective additional features : 8/10**

6.4.3 Tool Proficiency

Help file, manual document, tutorial and updates

Argumentative software provided a menu item for Help, including a help file directly accessible from the software, a *Check for Updates* feature which refers to the website, another link to the website home page and more information about Argumentative details such as version, licence and thanks. A manual is also available on the website as a helpful short movie for a quick introduction.

Documentation : 9/10

Errors and difficulties

An unhandled exception occurred such as illustrated on Fig 6.17 during a normal utilization of the system. This error seems to be a compilation error related to an unfounded resource. This exception is totally not supported by the software and appear every time the user try to change the type of a diagram component. This makes the software difficult to use if it is not possible to change the components type.

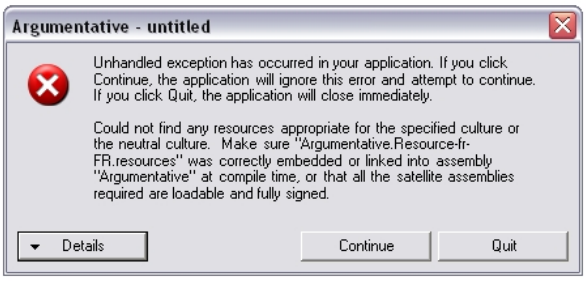


Figure 6.17: Argumentative - Unhandled exception met at the software utilization.

Furthermore, if user decides to quit the application, another pop-up window appears to ask him to save his diagram modifications. A cancel button is available on this window which should permit the user to return to his work without quit the application but instead of this, the application closes without saving anything and all unsaved data is lost. The errors management in the Argumentative system is bad.

Reliability rate : 5/10

Let's now look closer at the different structured visual forms offered by the Argumentative software application.

6.4.4 Argument Notation

The overview of the security requirements task about password authentication can be displayed in two different ways. The former, shown on Fig 6.18, is the colourful view, an aesthetic view of a top-down orientation diagram with all components included in colour boxes with shading and shadow options available. The latter is a more traditional display of the argument diagram illustrated by the Fig 6.19.

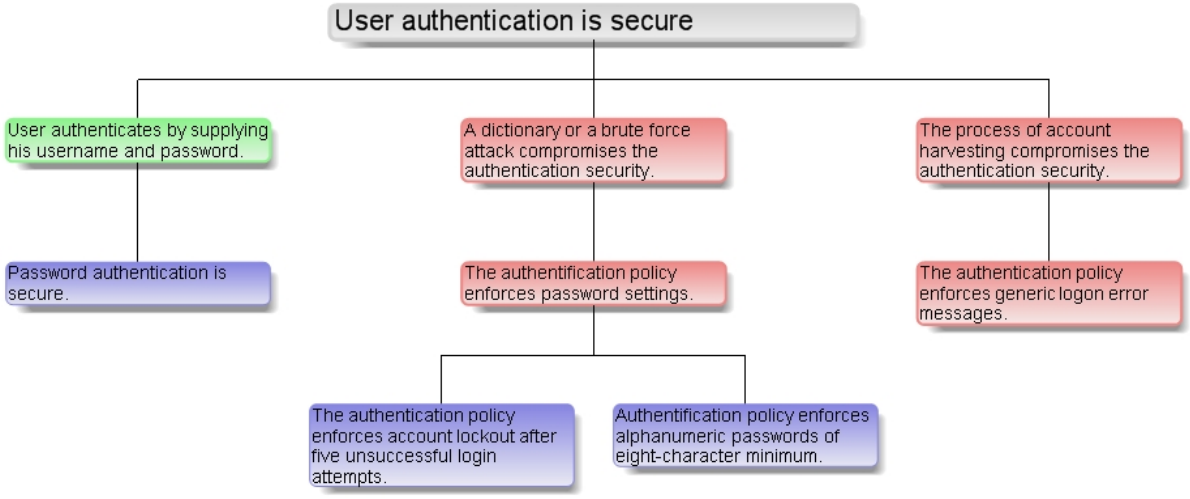
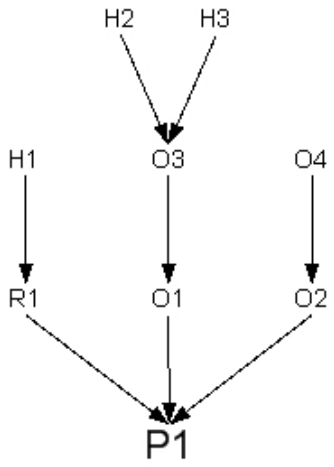


Figure 6.18: Argumentative - Representation of the colourful view of the task.

The second display corresponds to the R Notation option. It shows the argument in short form with reasons being numbered R1, R2,R3 etc, objections O1,O2... and helpers starting with H1. A legend showing the detail text for the elements is located just below the diagram. While the colourful diagram does not use arrows at the end of the straight lines, the R notation employs them.



P1: User authentication is secure
R1: User authenticates by supplying his username and password.
H1: Password authentication is secure.
O1: A dictionary or a brute force attack compromises the authentication security.
O3: The authentification policy enforces password settings.
H2: The authentication policy enforces account lockout after five unsuccessful login attempts.
H3: Authentification policy enforces alphanumeric passwords of eight-character minimum.
O2: The process of account harvesting compromises the authentication security.
O4: The authentication policy enforces generic logon error messages.

Figure 6.19: Argumentative - Representation of the R Notation view of the task.

Let's focus now specifically on the notations used in the colourful view illustrated on Fig 6.20.

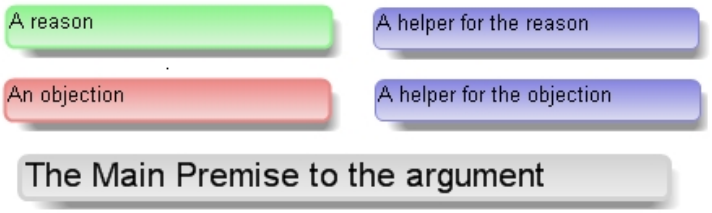


Figure 6.20: Argumentative - Visual components of the system.

The green background implies a reason to the premise it supports, the red one is at the opposite, an objection against the premise which it is related to. The blue background of boxes corresponds to both an helper for the reason or for the objection. The main premise to evaluate is represented in a box with a grey background. The symbol notations are very intuitive for the user and the argument reader. Only some black straight lines without any arrows are used to connect all boxes, no differentiation is

made according to the relationship between boxes.

Now, let's study the cognitive effectiveness of this set of notations using the five criteria. Table 6.8 gathers an appreciation rate and the explicative information for every criterion.

Table 6.8: Argumentative - Assessment of the visual language effectiveness.

Semiotic Clarity	$-$	The choice for the same symbol to represent an helper for the reason or for the objection is classified as a symbol overload of this symbol.
Perceptual Discriminability	$-$	The only differentiation available among the set of symbols is the background colour. This is not adequate enough to prevent interpretation errors or ambiguity to the comprehension of ideas underlying the diagram.
Perceptual Immediacy	$+$	The larger size of the police used in the box of the main premise to the argument gives intuitively the general idea that this premise is to be proved. Colours chosen are also good cognitive clues to indicate to the reader the good impression on reasons and objections affecting the value of the argument premise.
Visual Expressiveness	$--$	Only one visual variable is used, the colour. Even if colour is one of the most cognitively effective visual variable, this does not appear to be sufficient in a more complex diagram to be really efficient.
Graphic Parsimony	$++$	The visual vocabulary is cognitively manageable because of the diagram is composed of the same form in four different colours.

The result of the analysis of the cognitive effectiveness is not particularly good but nevertheless, with the utilization of the colour visual variable, the reader undergoes very little difficulty in understanding the ideas supported by the argument diagram.

6.4.5 Argument Ontology

The Argumentative application allows the user to move from prose to a visual structure by expressing all typical component : the main premise to the argument, a reason, a helper for the reason, an objection and a helper to the objection as the diagram shows that on Fig 6.16. These elements and the relationship between these elements express the underlying argument concisely. The definition of an argument changes from a series of premises to support a conclusion to a series of reasons and objections to support a premise where a premise itself may be a reason or objection in a subsequent argument.

The argument ontology is very simple and very close to the 'For' and 'Against' arguments in the user mind. In fact, the argument is composed of one main premise and a number of reasons and objections. The main premise is the conclusion of the argument. It is not called a conclusion because

each argument map may itself be used in a larger argument. Reason is a premise in favour of the main premise or conclusion while objection is a premise against the main premise or conclusion. The reasons and objections in turn may have reasons, objections and/or helpers against them.

This argument model allowed us to express a rebuttal of an objection. The biggest difference with others tools is the requirmeent to have only one main premise to argue. In our task, it does not raise an issue, but in more complex work, a different project will be necessary for each main claim.

For summarizing, we discovered in this tool a very efficient user interface, simple features and very good documentation. At the level of the argument conventions, the use of colour in notations makes a huge difference and the model is very clear and simple. Maybe too simple for a security argument design but still valid to express what we needed.

This leads us to the last tool assessment which highlights the key characteristics of the unique commercial product of the assessment: Rational 2.

6.5 Rationale

Rationale helps students deepen their understanding of good reasoning and critical thinking. But not only used in the educational field, Rationale is also one instance of an emerging category of software applications in the legal practice intended to improve legal reasoning.

6.5.1 User Interface

As this software is under a commercial license, the user is waiting for an high quality application. Once the user opened the Rationale system, he is captivated by the good looks interface of the application, as illustrated on Fig 6.21. Rationale provides indeed a very enjoyable user interface, tending to recall this of Microsoft Office 2007 software.

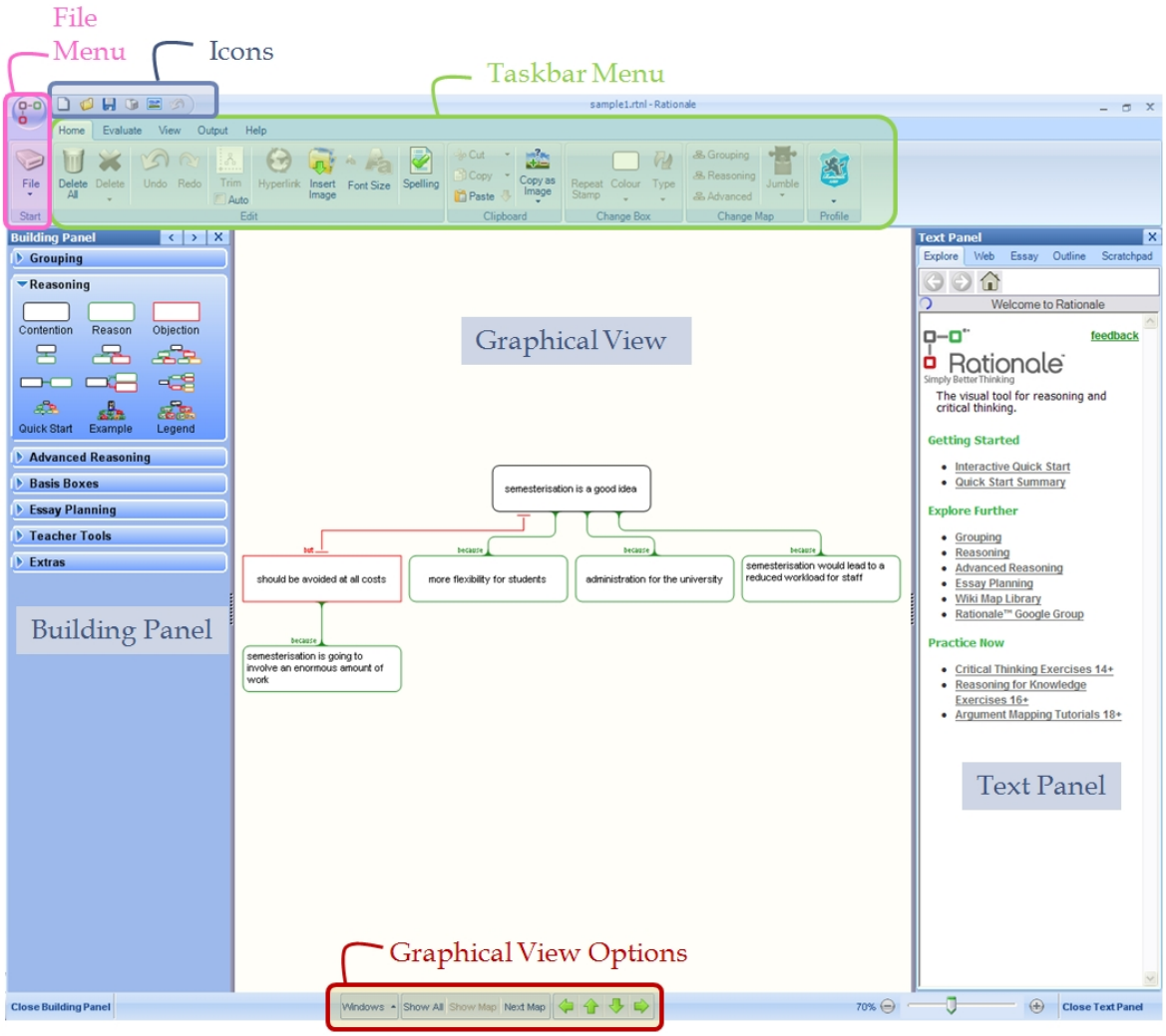


Figure 6.21: Rationale 2 - Screenshot of the graphical user interface.

User interface is composed of a Rationale specific big button giving access to the file menu, icons, a large taskbar menu, also called a ribbon with several tabs, different panels to support the diagram building and the user workspace equivalent to the graphical view. Let's review one by one these components which forming a very good overall result.

Menu and Icons

Rationale application offers a wide range of features through its menu. A specific menu is available for the file treatment and the ribbon, broken up into different tabs gathering similar features, gives access to various actions. For every item selected with the mouse, an message box appears with very complete information to inform the user on its utility.

Some icons with features on files are positioned at the top of the interface and others are listed under each tab of the ribbon. The size of these icons is large in order to be conspicuous. Unavailable features appear dimmed and cannot be selected. Furthermore, the Rationale menu is implemented to display visual effects on mouse-over such as button highlighting, etc. No information bar or status bar is visible in the Rationale system.

Menu : 10/10	**Icons : 10/10**

Building Panel and Graphical View

Besides a great menu, the Rationale software provides an effective process of diagram creation and a powerful map viewing. The building panel, on the left of the screen, gathers the different boxes from the various argument types and some templates to quickly design a diagram map. The easy way to build an argument consists in a click on the item in the building panel and dragging it to the workspace. The building panel is designed to allow the user to reveal the section with diagram symbols he needs during the diagram building and to collapse others in order to manage correctly the complexity of the display at the screen. The width size of the panel might be reduced or increased depending on the user needs. This panel can be closed and opened again via a button at the bottom of the interface.

The graphical view, also called the workspace, is the central panel of the high quality user interface. Basically, it looks like a blank page that user can fill and customize as he needs. Many features are implemented to help the user to manage correctly his workspace. For example, a button, placed at the bottom of the screen, helps navigate among the different open files. Using the slider on the bottom of the panel serves to zoom in and out. The zoom feature is also available by using the mouse wheel. Another specific button adjusts the zoom level and the workspace position in order to fit neatly the selected map into the window. A full screen view allows the user to effectively manage the complexity of argument maps. Several ways to move the window across the workspace are available, either by clicking on buttons, either by using arrow keys on the keyboard or alternatively, the user can pan by clicking on the workspace and dragging toward the wanted direction. Furthermore, an overview window positioned at the top left-hand corner of the workspace window reflects the position of the map on the workspace. The user can simply move the cursor into this overview window to move the map in accordance with the cursor movement. Drag and drop features are also very efficient to create and manipulate maps.

Graphical View : 10/10

Other

Predefined questions on user interface convenience reached an average score of 6.5/10 (See Appendix A.5).

To this result, one point is added for the pioneering possibility to customize the workspace. In fact, the user can choose an image in his files on the computer to display it on the background of the map.

Two other points are added for the text panel on the right of the interface, which can display the Rationale's built-in web browser. It allows to highlight text and drag and drop it onto the workspace. This panel is also used as a scratch-pad or when the user wants to plan and export essays with the specific feature which assists him to complete a structure and well-reasoned essay, letter, hypothesis or summary. Like the building panel, the text panel can also be closed and opened again via a button at the bottom of the interface.

User Interface Convenience : 9.3/10	**Other : +3**

Let's turn now toward Argumentative available features.

6.5.2 Features and Computational Services

Let's approach now the expected and additional features and computational services in the subsections thereafter.

Expected functions for argument mapping tools.

The Rationale system provides a large range of features available from the intuitive user interface. The Rationale file format (*.rtnl) is the only argument file format accepted in opening an argument map. Many output are available from one argument diagram such as vector and bitmap graphics and text outline formats. The printing of argument maps or the whole workspace is also easily possible. See Table 6.9 for a clear summary of expected features concerning argument files.

Table 6.9: Rationale 2 - Description of the file processing features.

Files Features :	
▷ Creation	· From a click in the building panel and dragging in direction of the workspace, · From the built-in web browser in the text panel, · From information copied onto the scratch-pad.
▷ Opening	· An argument file in the RTNL format, · Consideration of recent files.
▷ Saving	· As an argument file in the RTNL format, · As an image (emf, png, gif, jpg, bmp, tiff), · As an text (txt, rtf),
▷ Printing	A map or the workspace, with page preview and printing properties.

Various functions are available to manage diagram boxes and their content, such as editing the boxes' content by cutting, copying and pasting information, deleting boxes, duplicating boxes quickly. Another specific function consists to copy the map as an image. Rationale copies then the selected

box with or without the attached boxes below. Features which undo any operation and redo to get back seem also to be very useful in this software. The software offers the possibility to edit some parameters characterizing the user preferences such as the font size of boxes content, the background picture, etc. The type of argument boxes can be changed, as the whole argument map can also be adjusted in another sort of maps.

Expected functions : 9/10

Additional features

Some additional features very useful are offered to the user through actions available. Among them, some should be mentioned such as an additional taskbar using for finding text in the argument map, a button to flip the map orientation, a spelling checker, an hyperlink editor or even a ready to go activity which consists in jumbling the map and swapping jumble puzzles with someone else. An additional export feature helps create and email Rationale work bundle through an easy process. Teachers could find also some functions to aid feedback to their student, such as sticky notes, feedback notes,

Another wizard is available for students to plan and export essays. By following the instructions to create a plan, the user can watch the essay taking shape in the text panel. When the map is complete, an export button starts the essay generation. Then the essay opens as a word processing document.

Effective additional features : 8/10

6.5.3 Tool Proficiency

Help file, manual document, tutorial and updates

The Austhink institution provides extensive and accurate documentation for the Rationale application. A local help file is available by a simple click from the help tab of the ribbon and many pages on the Rationale website are directly accessible from other icons such as a link for opening the Rationale's home page, the technical support, the user feedback survey, the forum, or learning materials including guides for educators and professionals.

Furthermore, the website provides a video where the people behind Rationale tell the story of its development and how it improves critical thinking skills. Published papers and articles are listed there too. Some map examples packs can be downloaded and the website includes also a shared online map library to find many resources.

Another button in the Help tab of the application permits to check if there is a later version of the Rationale software available.

Documentation : 10/10

Errors and difficulties

A manipulation of the software resulted once in a full system crash without any known reason and without any error messages nor argument saving.

One of the main technical difficulties encountered is the inability to combine different map formats.

Reliability rate : 7/10

6.5.4 Argument Notation

Fig 6.22 illustrates the argument diagram in the advanced reasoning style achieved with the Rationale system.

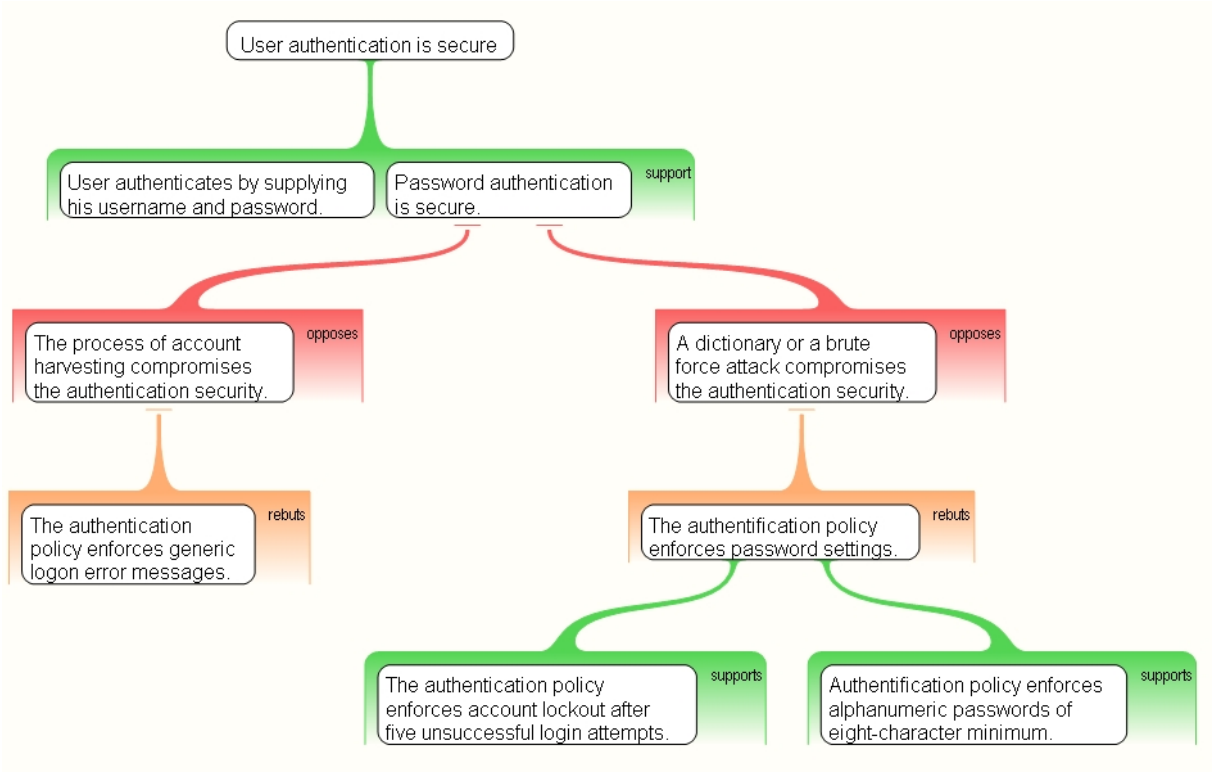


Figure 6.22: Rationale 2 - Representation of the argument task.

The structure created with the enhancement of visual diagram is under a pyramidal form. On Fig 6.22, the first reason supporting the contention is a composed reason. This is represented by one reason inside the box of the other. This kind of representation to express the support to another statement is met for the first time in the assessment.

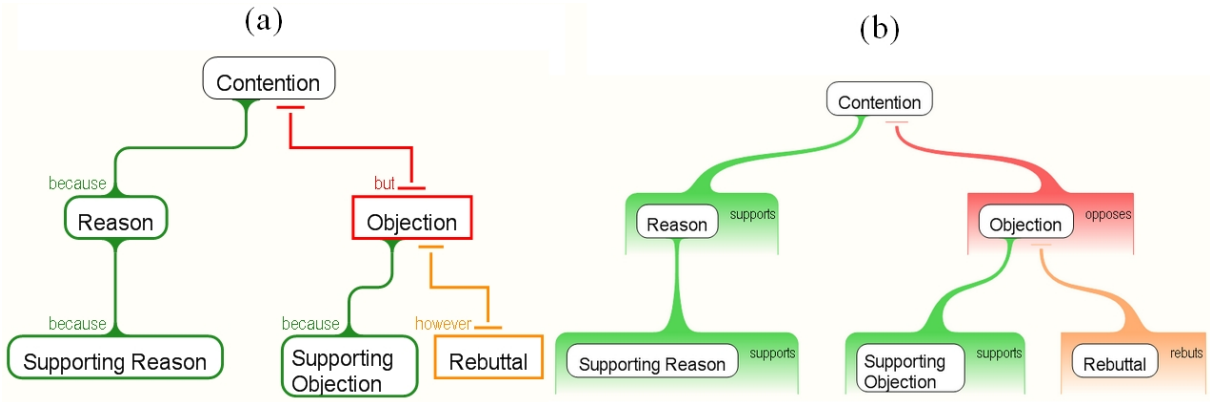


Figure 6.23: Rationale 2 - Visual components of the system.

Let's now look closer at both notations syntax of reasoning style on the part (a) of Fig 6.23 and advanced reasoning style on the (b) part.

Colour and some textual labels on the map provide a visual indication of the evaluation of the claims and basis boxes. An option to evaluate the claims and the strength of the reason as a whole is provided in advanced reasoning maps.

Table 6.10 gives detail about the assessment of cognitive effectiveness of Rationale argument notations.

Table 6.10: Rationale 2 - Assessment of the visual language effectiveness.

Semiotic Clarity	−	Reason, Supporting Reason and Supporting Objections concepts have just the same notations on the map, that can imply confusion to the reader because of this symbol overload or the symbol deficit for these concepts in both styles.
Perceptual Discriminability	+	Discriminability between concepts is possible thanks to colours and boxes corners with rounded corners representing supporting statement while straight corners describe objections or refutations. Relationship links are also different from their nature which allows the user or the reader to more easily differentiate premises.
Perceptual Immediacy	+	Red, green and orange colours are chosen to directly express ideas under the concept.
Visual Expressiveness	++	Many visual variables are used : Colour, horizontal and vertical position, shape of elements
Graphic Parsimony	++	Only three different relationships and four different nodes notations are needed to express basic argument.

The analysis leads us to still highlight the preponderance of the visual variable colour and its importance to express ideas and be understanded easily.

6.5.5 Argument Ontology

The ontology underlying this visualizing argumentation software is composed of a contention equal to the conclusion to prove, reasons and supporting reasons which support the contention, objections and supporting objections which reject the contention exactly as we already found in the Argumentative software ontology. But in addition, this ontology takes into account the refutation concept which refutes an objection with which it is connected. This new separation between objection and refutation is very beneficial for the reader.

Obviously, the specific aspects related to a commercial product such as a very user-friendly and intuitive interface or very detailed and complete documentation are dominant in the results. This application gives the best results among other tools which is not very surprising since it is a commercial product.

To complete the statement of results in this chapter, lets enumerate observations on the set of results.

6.6 Gathering of Results for a Quick Comparison of Tools

After having developed every tool in the first part of this chapter, we are going now to bring together the scores of participating tools In order to allow quantitative comparisons. Lets start by the results of the user interface convenience whose results were in the annex A, and then we examine the results obtained on the tool experience, the first of the two different approaches of the assessment. Finally, we end this section by analysing the different results about the cognitive effectiveness of the set of notations.

These results should not be considered as evidence because the process of the method used to collect the results is subject to a margin of subjectivity because of the small sample size. However, these assessment results allow us to consider some interesting findings on visualization argumentation tools emerged from the examination.

6.6.1 User Interface Convenience

The results displayed in the Table 6.11 represent our first data which we can deduce some statements.

Table 6.11: Comparison between the five tools on user interface convenience.

Criteria	Araucaria	Athena	ArguMed	Argumentative	Rationale
Information Accuracy	9	3	4	8	10
Balanced Interface	7	7	8	10	10
Interface Intuitiveness	5	9	6	8	9
Visual Rendering	7	5	8	9	10
Visual Attractiveness	5	5	5	10	9
Diagram Scalability	8	7	8	9	9
Direct Manipulation	7	9	7	8	8
Zoom Magnification	7	7		10	9
Convenient Performance	3	8	5	9	10
Average Score	**6.5**	**6.5**	**6**	**9**	**9.3**

Firstly, the best average score is this of the Rationale software. Commercial software must offer unrivalled quality with open source software to the user who buys a license of the product. The commercial software is closely followed by the open source software Argumentative which appear to be very on the point of view of the user interface. The three other tools are almost equal with an average score of around 6.5/10. The worst performance in terms of user interface convenience is attributed to ArguMed. This can be explained by the major interest of this tool for the argumentation conventions.

On this table, we can also observe the lack of a zoom feature for one among the tools, but this can be explained by the representation of the argument as a tree and not under the form of a diagram which limits the extent that the argument can take.

The first line of the table shows us that an effort is to be made for two of the five tools in terms of accurate information contained in the user interface. Athena and ArguMed are probably more focused

on research and prototype which may reduce the necessity to always provide complete information to a novice user.

The visual attractiveness is not the best developed property for non commercial software. That can be attributed to a lack of resources or interest of developers who focus primarily on features of the tool. If the objective of the software to create and manipulate argument is met then there is no real need to improve the software workspace with visual enhancement. Non commercial software is not designed to be flamboyant, but rather to be primarily used effectively.

In a general way, the complexity of large scale arguments is well managed through implemented mechanisms in every analysed tool and interfaces are balanced correctly between the set of available panels.

After describing the results mainly centered on the user interface convenience, lets examine the more global results on the tool experience.

6.7 Tool Experience

Unsurprisingly, the commercial software tool is at the top of results as showed in Table 6.12. An excellent tool experience for a commercial software of this kind of tool is essential to make the difference.

Table 6.12: Comparison between the five tools on the tool experience.

	Araucaria	**Athena**	**ArguMed**	**Argumentative**	**Rationale**
User Interface					
Menu	9	8	5	8	10
Toolbar Icons	9	8	7	9	10
Status Bar	8		7	8	
Graphical View	5	8	7	9	10
Other	+2	+1		+2	+3
Features					
Expected Functions	8	8	8	9	9
Additional Functions	7	8	7	8	8
Tool Proficiency					
Documentation	8	6	7	9	10
Reliability Rate	6	8	9	5	7

The lack of a status bar for two among the tools is rather unfortunate because the user likes having control over the tool and stay informed of the result of his operations. The Table 6.11 also highlights

a minimal difference concerning data results about the expected and additional functions.

The reliability rate is too subjective, so we do not consider it in this review, however the documentation provided to correctly use each tool is an essential point of the tool experience. Athena provides documentation just sufficient for the novice user needs.

The last comparison that we will discuss in the next section deals with cognitive effectiveness of argumentation notations.

6.7.1 Cognitive effectiveness

Concerning argumentation notations, we studied the cognitive effectiveness of the visual conventions used by all these tools in order to highlight the main visual qualities. The reader can see in Table 6.13 the collection of estimation data that we describe briefly below the table.

Table 6.13: Comparison between the five tools on the cognitive effectiveness of argumentation notations.

	Araucaria	**Athena**	**ArguMed**	**Argumentative**	**Rationale**
Semiotic Clarity	−	−	+	−	−
Perceptual Discriminability	+	+	++	−	+
Perceptual Immediacy	++	−	++	+	+
Visual Expressiveness	++	−−	++	−−	++
Graphic Parsimony	++	+	++	++	++

The semiotic clarity is wrong for most tools. The correspondence one-to-one between a visual symbol and an underlying concept is rarely respected. The reader can then misunderstand some relationships between same notations or can misinterpret the ideas conveyed. In some cases, the reader needs to read the informational content to understand the link between ideas while the presupposed idea of an argument map is to be able to directly differentiate the different parts of the argument and their relationships.

During the assessment, we often find that the visual variable colour is the most used variables among all. Indeed, this variable is the most efficient so its use is more important than others, but it would be valuable to extent the differences between symbols to others visual variables which are never took into account such as the shape or the bright of components.

Some software tools use display rules to get a better rendering of their diagram and in most of these cases, the horizontal and vertical position are very useful to the perceptual immediacy.

The graphical parsimony is generally very good because argumentation ontology is simple and does not require a too complex visual vocabulary.

6.8 Summary

In this chapter, we examined in turn each of the visualization argument tools by carefully following criteria of the assessment. The next part includes a discussion considering the essential information discovered in these results.

Part III

Discussion

Chapter 7

The Analysis of the Results and Discussion

This chapter summarizes by criteria the results obtained in the assessment with the aim of highlighting the key characteristics of a better computer-supported argumentation visualization tool. It looks also at the issues which are raised by the approach of in which these software tools are used to formalize the security requirements in the form of the visual argument maps. Finally this chapter ends with a review of the process which led us to construction of this assessment such as the criteria and the choice of the selected tools.

7.1 Discussion of the Assessment Results

In order to properly analyse this assessment, it is essential in this section to discuss what has attracted our attention for each evaluated criterion.

7.1.1 Recommendations on the User Interface

The user interface is the main interaction tool between the argumentation software and the user. A good intuitiveness of a visual argumentation software seems to be an essential quality for this kind of software because it improves significantly users' experience with the software. The main requirement of the user is to keep control of the tool and of his work. The mouse device is, without any doubt, a very useful pointing device for transmission of commands of the user to the software. It is, therefore, entirely desired that features like point-and-click or drag-and-drop are available in order to enhance the exchange and synergy between the human and the software.

Besides this, the visual rendering of the diagram arrangement following some positioning rules is a visual advantage for many tools but can be also considered as a limitation, because it may lead to a feeling of loss of control by the user. Indeed, if allowed, arrangement of the different boxes by the user would be a significant improvement in performance of customization of the graphical environment.

Menu tooltips, providing clear and relevant additional information, also increase usability of the software. Suitable icons improve the easiness with which the user executes the available actions. The use of hotkeys allows the user to work more efficiently and to save time.

Another concern regarding the user control is about the feedback provided to the user after some operations like saving a document have been executed. An informational panel like the status bar at the bottom of the screen is useful for notifications about task completions. Displaying of error messages with actions to be taken when disruption occurs allows for management of an inappropriate behaviour. An error management mechanism can be used to capture all relevant error information and then report it immediatly online in order to contribute to improvements of the software.

Customization of the software interface can be perceived differently from one user to another. It may be very useful for some users who get used to the software more easily in this way, or it may be identified as totally useless for other users who are not concerned with software aesthetics of the software, but only with functionality of the software.

On the other hand, customization of the boxes and appearance of the whole argument is something more valuable because in most cases, the argumentation map is being designed to be read by someone else than the arguer. The argument map is then the arguer's idea which he wants to convey. In this case, a good graphical appearance of the argument has a higher value to the target consumer.

7.1.2 Recommendations on the Available Features

Our recommendations about the functions and features provided by visualization argumention mapping tools are mainly focused around the general usage of these programs, although sometimes we look more specifically at features related to some specific fields of usage such as the education domain, where availability of the tutor function is highly desire. At the level of argument editing, some functions as copy, paste or undo, redo appear as significant for a good software tool.

A specific concern that we have noticed during this assessment is the wide range of file formats used for storage of the argumentation maps. For each tool, a specific file format has been created to correspond to a specific information and storage needs. A standardisation of the argument file format would lead to easier sharing of the users' map.

Besides this, the work realized with the visualization argumentation map tool is a part of a larger process in which people want to communicate an argument structure to prove a certain claim to a specific audience in a clear, effective and coherent way. The ability to export the result of the work, to save the argument in a picture form, to create a customized report appear as essential operations required by this kind of software. We have also observed that another useful function also observed is the option to generate a slideshow containing all subsequent steps in creation of the argumentation map; it provides guidance for the reader through the different boxes.

Now, if we consider the evaluation function of the argument, which is not provided by all tools we assessed, it appears that, this function supplies a significant advantage, because it supports reasoning of the mind and allows for changing of some of the input values and for direct observation of the result of the evaluation calculation. User can then re-edit the argument components until the evaluation process gives the expected result. This function can also help to improve critical thinking by highlighting the necessary elements the process needs to take into account. Having said that, the evaluation carried out by the evaluation process or in the users' mind behind his computer, or by a discussion group are not always the same. This is why it is preferable not to assign the value "true" or "false" to the

different assumptions of the argument. Furthermore, these values can be subjective or even wrong, while it is only the declaration of the argument structure which leaves the door open for discussion. Evaluation is very practical in terms of decision making, but in some cases, the evaluation process cannot reach a decision. These situations are indeterminable with only elements of them known. In our assessment, the evaluation process available in some software tools was not used because the data it requires have not been specified as it had not been the subject of our work.

7.1.3 Recommendations on Software Proficiency

The availability of a tutorial divided in well-defined steps, helps more people enjoy the software and perform the first steps with the software. It should not be too long or too short either, but it should give a perfectly suitable overview of the major functions and can also refer to a help file accessible from the software for less critical features. The tutorial serves also to draw attention to the additional features available in the tool.

The help file included in the software and accessible via a menu item is also essential for the user learnability since it contributes to inform the user. The content of this file gathers clear definition of terms used in the interface, but also a complete and accurate description of all the functions and options available with the software package. This file can be structured such as the menu items to allow the user to identify easily desired sections in the same way he learned to navigate between menu items in the software. A list of the Frequently Asked Questions can also be inserted in the help file to resolve first troubles that might arise in some user experiences.

The software website offers generally additional information and more up-to-date information. A link to the website from the software is highly required. However, redirecting the user to the website for simple information is not always the best solution even if, nowadays, workstations are more and more connected to the Internet. That totally breaks the current work of the user with the software, by displaying new information at the screen, not always relevant with his current research. The redirection requires of the user a better ability to sort information and more concentration in order to get back at the software to continue the user's current task.

Naturally, beside this negative point, the website is an essential resource for the software user. It allows providing argument map examples already built with the software. The website has also a role for new listing updates or simply to gather users of the software in a newsgroup or a forum where they can ask questions and others can respond. Website is also the location where user can find reports, publications or promotional articles about the software tool. These give a better knowledge of the specific domain in which the software is inserted.

A last recommendation related to the software proficiency is about the errors management. The design and implementation of software is rarely perfect, mistakes can occur and induce errors during the software utilization by the user. What is essential in this case is that the user does not lose his current task. The best for the user is that the error be managed in a way that the user be able to return, without any difficulties, to his earlier task that he was going to do before the error occurred.

7.1.4 Recommendations on the Argumentation Notations

Notations of the argument are the medium to convey efficiently the information. To discuss and make some recommandations, we take the place of the reader which attempts to understand the argumentation map without any other explanation. Colour is the first thing the human will decode. As we already said, colour is one of the most cognitively efficient visual variables, so the effect on the human mind is immediate. The colour would be used adequately. Too few or too much colour used in the diagram loses the valuable effect of this visual variable. The discriminability between a red box and a green one is almost immediate, but the reader will not perceive an immediate difference if fifteen different colours are used on the diagram.

Furthermore, there exists a relationship between the visual notations and the underlying concepts. In cases we have met in the assessment the visual vocabulary was manageable because of the argumentation models we met are simple enough, but if the number of concepts for representing the diagram is about fifteen, twenty or more, the visual notations must be specified properly to correspond to every theoretical concept. This implies the utilization of a larger combination of the visual variables for keeping a minimal rate of complexity.

We may sometimes forget the role that the relationships links play in the diagram. Among the five tools analysed, some do not take into account the importance the links have in the cognitive comprehension of the reader. Caring the notation links between the boxes is also improving the visual rendering of the diagram. Other visual elements are used inside the boxes. They may suggests differences between two boxes of the same notation.

As a good security manager should have the mind of an attacker to understand the system failures, the notations designer must be put in place of the reader to find his inspiration.

7.1.5 Recommendations on the Argumentation Ontology

In different tools we have assessed, we noticed that the vocabulary used for a same concept can be very different following the different argumentation model styles the software implements. For example, the thesis to be supported is called the contention in Rationale, the main premise in Argumentative, the assumption or the issue in ArguMed following the specific evaluation. So for a user which could use several among these software tool, ontology concepts could seem hard to correctly interprete only on the basis of the noun.

In every software tool, the argumentation ontology we studied was not too complicated. A main premise is supported by assumptions that we named reasons and helpers of the reasons. Objections refutes the main premise or others assumptions. In somes cases the objections of another objection is considered as another concept, the refutation. We found mainly the basic concepts of the argumentation theory.

We refer the reader to the next section for a more complete explanation about the ontology for expressing an argument such as a security requirement.

7.1.6 Summary

The ideal visualization argument mapping tool would have the characteristics described here above in detail. It would include an intuitive user interface allowing easy manipulation of the table and ensuring adequate information for full use of the software. Furthermore, it would provide the expected features such as the creation of a new project, editing it, saving it. Exportation or printing of the current task is an indisputable advantage. At the diagram level, expected features include adding or editing types of diagram components, their content and the ability to delete items become obsolete. A good documentation is also an essential quality of software and the usage of the software is related to the confidence that the user will have in the software so a better reliability rate implies a better level of confidence.

If we now consider the argument map created by an ideal tool, it will highly dependent of the argument style used. But we can say that visual symbols should be clearly defined and distinctly linked to underlying concepts belonging to the model theory. Moreover, the visual vocabulary of this model would be manageable. And finally, we also denote among required qualities an efficient utilization of the visual variables, among them the colour, which is, as we previously discussed, the easiest visual variable to cognitively distinguish.

In addition to the necessary basis, an ideal tool could propose the creation of templates files to create personalized model of arguments. A spell checker is undeniably useful for establishing the user credibility. We can also add to these additional features, the availability of a tutor mode, relevant for a utilization of the software in philosophical course for example. A tutor mode includes functions specially designed for the teacher to read, comment or revise the tasks of students. The tutor mode was met with many tools during the assessment.

7.2 Focus on the Place of Security Requirements

In this section, the question we want to ask is: Is it relevant that a complete analysis of security requirements be summarized in a confrontation between objections and supporting premises for determining the security risk result. In order to attempt to answer this question, let us explore the advantages and disadvantages of creating security arguments with CSAV software tools.

The considered task in this assessment about the password authentication includes some security concepts as threats and vulnerabilities or potential attacks that we designed such as rebuttals on the password authentication feature. Then we described some security mechanisms implemented to reduce or eliminate the vulnerability and fulfil the security requirement that the authentication is secure. The first advantage that we highlighted here is the possibility to express the security concepts with the argumentation model components implemented in each tool. Obviously, all the tools are implemented differently, so some restrictions of use was encountered, but we can declare here that a simple security requirement can be expressed with this kind of tools within the limits of complexity.

Another advantage directly related with the security domain is the presence of security concepts which attack each other such as a vulnerability attacks a system's asset, or a security mechanism attacks the threat due to an attacker profile. The support relationship is also present among the security concept. For example, a security requirement supports the security policy or the threat supports the real attack. Many relationships in the security domain can be summarized by the links *support* and *oppose* of the argumentation conventions.

The third advantage is that each statement in the argument is a declarative sentence expressed in natural language. This implies that if something is missing in the argumentation model, it can be explained in natural language inside a premise statement in order to help the reader understand the arguer conception. The use of natural language is very practical to address specific weaknesses related to the expression of the security requirements.

Regarding now the disadvantages, the main trouble is at the level of the ontology or the argument model's own style used to represent the security system. Indeed the conception of security requirements is asking for a more specific argument model than the expression of objection and support items. The security engineer would like to distinguish the security concepts in the theory but also in practice on the argument map without resorting to tricks textual. He would like to declare a specific statement being an asset threatened by a vulnerability either existing or potential which can be exploited during an attack initiated by a specific attacker profile, etc.

In many cases, this problem seems to be resolved by the creation of a specific template in some software where this feature is available, but the main difficulty would be then to correctly convey the argument ideas to the reader. In fact, the reading of an argument map representing some concepts of security requirements may be difficult due to the lack of cognitive effectiveness among the additional notations which are often very similar. The visual distance is reduced between the different elements of a designed template.

Finally, the last disadvantages we still notice is related to software with an evaluation function. These tools are implemented to attribute a result value to the main claim. This result value corresponds simply to true, false or indeterminable, while for a security requirements map, the resulting risk is a more complex calculation method.

To quickly conclude this section, we summarize by saying that designing security requirements and other security concepts with a computer-supported argument visualization tool is possible, because security implies various objection/support relationships and these tools works on basis of premise statement written in natural language, this let a certain freedom rate to express additional information needed in the security domain. So it is possible, but it leads the user to have to adapt according to circumstances and needs of expression and it makes the reading of the argument map more difficult and reduces the cognitive understanding of ideas.

7.3 Assessment Review

Here we are arrived at the end of the discussion about the assessment's results. After such hard work to reach the conclusion of the assessment, it may seem interesting and useful to discuss about what have been the main difficulties we encountered during the entire process. In fact, three major difficulties seem interesting to report here: choosing tools, defining criteria and determining the correct place to security requirements.

Regarding the selection of tools, a dozen different software applications were initially selected to be part of the assessment. We eliminated many among the list as we went along for some reasons.

For the Carneades software[1], no user manual file exists yet and the implementation of the tool does not correspond with publications about the system. After an exchange of email with the software contact, we learned that the software was currently ported to the Java Abstract Machine using the

[1] Available at `http://carneades.berlios.de`

Clojure programming language. It was therefore difficult to assess the software under these conditions, even if publication ideas seemed very interesting.

The Belvedere software[2] has never worked or started under two different configurations of test, so we removed it from our list.

The Reason!Able software[3] by Austhink is the predecessor of Rationale and is no more available now.

The bCisive software[4] is another application produced by Austhink but more oriented through collaboration for decision-making, planning and team-problem solving, so we only selected Rationale more focussed on argumentation to assess a commercial product of Austhink.

The commercial QuestMap software[5] which *originated from the pioneering hypertext system built by Jeff Conklin*[6] was also no more available on the Internet.

The tool that would probably have its place in a larger assessment on tools helping manage the connections between information and ideas, is the offspring of the QuestMap software, called Compendium[7], whose development is based at the Open University (in collaboration with Verizon, NASA and the Cognexus Institute). This was also removed from the list because it encompasses much more collaboration operations and others available operations than what we needed for this assessment, but it is a very good tool for larger collaborating activities than argumentation.

Finally, the last tool which we had to give up is Visual Network Rating Methodology[8], developed by the Naval Research Lab of Washington. VNRM is *a toolset and language for developing and evaluating a map of an argument that mission-critical information is adequately protected by a system in its larger operational environment*[9], but is not available to download.

Regarding criteria we defined to assess the selected tools, the first idea was to specify them from some worldwide standards written by the specialized body composed of The International Electrotechnical Commission (IEC) and the International Organization for Standardization (ISO)

We also thought find more information in the Guide to the Software Engineering Body of Knowledge [4], but it was not exactly what we looked for to consider the different angles, so we quickly gave up this orientation. Actually, SWEBOK is the result of the IEEE Computer Society which had established for the first time a baseline for the body of knowledge for the field of software engineering, and the work supports the Society's responsibility to promote the advancement of both theory and practice in this field.

Other guidelines can be very interesting such as the Human Interface Guidelines (HIG) which is a document about software development including a set of recommendations for application developers in order to improve the experience for the users by making application interfaces more intuitive, learnable and consistent.

But it is finally by reading [44] that we divided the assessment into our five different parts. In this publication, the author presents five key characteristics for modeling discourse as hypermedia networks: the discourse ontology, one or more notations, an intuitive user interface, computational services and the literacy and fluency of the user. The article gives an instantiation of their key charac-

[2]Website is at `http://lilt.ics.hawaii.edu/lilt/software/belvedere`

[3]See at `http://rationale.austhink.com/reasonable`

[4]More information at `http://www.bcisiveonline.com/`

[5]A description is available at `http://compendium.open.ac.uk/institute/tools/tools.htm#questmap`

[6]`http://compendium.open.ac.uk/institute/tools/tools.htm`

[7]Available at `http://compendium.open.ac.uk/institute/about.htm`

[8]Additional information can be found at `http://chacs.nrl.navy.mil/projects/VisualNRM/`

[9]`http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA378599`

teristics for two of the hypermedia discourse concept: Compendium and ClaiMaker. Our five criteria were grouped mainly on the first four key characteristics, which resulted in a fifth for our assessment by derivation of ideas. We finally produced a clear definition of our five criteria in this way.

Other aspects or approaches of these software applications could have been taken into account or examined in more detail such as the user experiment, the time needed to achieve the task, the number of needed attempts, etc. but we have only included those particular aspects which already imply together a complete assessment.

7.4 Summary

In this penultimate chapter, we presented a large discussion about all advantages and desadvantages we met during the assessment process with argumentation software tools. We discussed one by one the criteria of the assessment in order to elaborate some specific recommendations. The list that we presented is far from being exhaustive or treating all parts of aspects without omission. After this part, we highlighted some troubles met with the design of an argument such as a security requirement. And finally, we gave our critics of this assessment process.

Lets finally enter in the last chapter in which we present our conclusions and we explore what could be the continuation of this work.

Part IV

Conclusion

Chapter 8

Conclusion and Future Work

After the reading of this thesis, what do we conclude? Lets try to answer to this question by reconsidering our thesis objectives.

Firstly, lets turn a moment to look at our global objective which was to be able to use an argumentation visualization software tool to design a specific argument from a set of security requirements. This first objective has been reached because we have succeeded to build a security argument with each software tool included in the assessment process. But, the concrete task we defined to the test is only a simple example of what security requirements represent really. The real needs of security manager are much more complex in terms of the requirements they design in very secure systems.

The difference between our simple example and the real problems is huge. However, our example allowed us to determine the current advantages and disadvantages of using this specific kind of software tools for designing a visual diagram considering the expression of security concepts. The main advantage is that many relationships in the security context can be resumed by an argumentation relationship such as support or refute. This main property is the reason that we succeeded to build the argument task with all of these software tools. We also showed that the argumentation model seems to be insufficient to express all the semantic of the key concepts in security requirements. It is the reason why future work on the subject should be more oriented to the model sharing the security concepts.

Among the other objectives of this thesis, we had planned to carry out a complete review of a selection of visualization argumentation software tools. The assessment process resulting with a large amount of results which have been analysed presents a very complete overview on many approaches of these five tools. The selection of these tools has also been discussed in the penultimate chapter. See Section 7.3 for information.

We think that with this thesis which represents a basis of the evaluation, it might be easier and simpler for completing other exhaustive assessment on other software tools close of the visualization argumentation concept, such as for example Compendium that we had eliminated of our assessment list because it is more centered on collaborative work of argumentation. A future work in this orientation could be to extend the selection of tools and to submit the analysis description to a group of different users. That could avoid the subjective results that it is possible to find in this thesis because of the size of the sample used here.

Finally, our third and last objective was to determine keys characteristics for the best argument mapping tool. The perfect argumentation software tool does not exist, but we have found some relevant characteristics that could improve the result of a developing work on the subject. Firstly, the ideal visualization argumentation software tool includes an intuitive, user-friendly and easily manipulable user interface. Its user interface provides all the necessary information to users and what is not found on the interface is easily accessible in a help file or a manual. Available actions and different features must meet all the needs of users while the software remains under control. Obviously, these qualities are abstract enough to developed many very good tools which might be more pleasing to some users than others depending on his or her preferences.

Finally the main objective of these visualization argumentation software tools is to assist the user in his or her job. In our world that seems us today maybe so closed behind our computer screen, it is essential to note that our job, the job of a computer scientist is primarily to facilitate the life, business, research work, learning, etc. of other users.

Bibliography

[1] Testing guide introduction - OWASP. http://www.owasp.org/index.php/Testing_Guide_Introduction.

[2] Argumentative Manual. http://argumentative.sourceforge.net/LinkedDocuments/Argumentative%20Manual%200.5.pdf, 2008.

[3] Getting started with Rationale. http://assets.rationale.austhink.com/pdf/gettingstartedguide.pdf, 2008.

[4] Alain Abran, James W. Moore, Pierre Bourque, and Robert Dupuis. The guide to the software engineering body of knowledge, 2004.

[5] Aristotle. Rhetoric. http://philosophy.eserver.org/aristotle/rhetoric.txt, -350. Translated by W. Rhys Roberts.

[6] Aristotle. *On Rhetoric : A Theory of Civic Discourse*. Oxford University Press, 1991. Newly Translated, with Introduction, Notes, and Appendices by George A. Kennedy.

[7] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[8] Simon Buckingham Shum. The Roots of Computer Supported Argument Visualization. pages 3–24, 2003.

[9] Toni Buzan. *Making the Most of your Mind*. Pan Books, 1977.

[10] Chad S. Carr. Using computer supported argument visualization to teach legal argumentation. pages 75–96, 2003.

[11] Carlos Chesnevar, Jarred McGinnis, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, and Steven Willmott. Towards an argument interchange format, 2006.

[12] T. De Jong, S. Ainsworth, M. Dobson, A. van der Hulst, J. Levonen, P. Reimann, J. A. Sime, M. W. Van Someren, H. Spada, and J. Swaak. Acquiring knowledge in science and mathematics: the use of multiple representations in technology-based learning environments. pages 9–40, 1998.

[13] Merriam-Webster Online Dictionary. Evidence. http://www.merriam-webster.com/dictionary/evidence, 2010. Retrieved April 14, 2010.

[14] Frans. H. Eemeren, R. Grootendorst, and F. S. Henkemans, editors. *Fundamentals of argumentation theory : A handbook of historical backgrounds and contemporary developments*. Lawrence Erlbaum associates, Mahwah, New Jersey, 1996.

[15] Sonja K. Foss, Karen A. Foss, and Robert Trapp. *Contemporary Perspectives on Rhetoric, (Third Edition)*. Waveland Press, Long Grove, Illinois, 2002.

[16] Tim Van Gelder. How to improve critical thinking using educational technology. In *Biomedical Multimedia Uni, The University of Melbourne*, pages 539–548, 2001.

[17] Michael Gerdes. An exploration of global perceptions of security and privacy. *Information Systems Control Journal*, 6, 2002.

[18] T. F Gordon. Visualizing Carneades argument graphs. *Law, Probability and Risk*, 2007.

[19] Yehouda Harpaz. What is "evidence". `http://www.human-brain.org/evidence.html`, 2000.

[20] Maralee Harrell. Using argument diagramming software in the classroom. *Teaching Philosophy*, 28, 2005.

[21] International Electrotechnical Commission International Organization for Standardization. International standard iso/iec 9126. information technology – software product evaluation – quality characteristics and guidelines for their use, 1991.

[22] International Electrotechnical Commission International Organization for Standardization. International standard iso/iec 27001. information technology – security techniques – information security management systems – requirements, 2005.

[23] A. Laghos and P. Zaphiris. Frameworks for analysing Computer-Mediated-Communication in e-Learning. *Lambropoulos, Niki; Zaphiris, Panayiotis (Hg.): User-centered design of online learning communities. Hershey, PA: Information Science Pub, S*, pages 238–264, 2007.

[24] Richard Long. The role of audience in Chaim Perelman's new rhetoric. http://www.jacweb.org/Archived_volumes/Text_articles/V4_Long.htm.

[25] L. Thorne McCarty. Some arguments about legal arguments. In *ICAIL '97: Proceedings of the 6th international conference on Artificial intelligence and law*, pages 215–224, New York, NY, USA, 1997. ACM.

[26] Sean McCreary. Business security information : Addressing the most important business security issues. http://www.businesssecurityinformation.com/2010/06/business-security-issues/.

[27] Nancy Mead. The common criteria. 2007.

[28] P. S. Meier. Mind-mapping: a tool for eliciting and representing knowledge held by diverse informants. *Social Research Update*, 52:1–4, 2007.

[29] K. L Mills. Computer-supported cooperative work. 2003.

[30] H. Mintzberg. *The structuring of organizations: A synthesis of the research*. Prentice-Hall, 1979.

[31] Daniel Moody and Jos van Hillegersberg. Evaluating the visual syntax of UML : An analysis of the cognitive effectiveness of the UML family of diagrams. 5452:16–34, 2008.

[32] Daniel L. Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35:756–779, 2009.

[33] Philip Moustakis. Missing: A death penalty debate. *The New York Times*, February 2000.

[34] Denise O'Berry. Small business security – Your survival depends on it. `http://www.allbusiness.com/operations/3872802-1.html`, 2006. Retrieved July 8, 2010.

[35] National Institute of Standards and Technology Gaithersburg. Announcing the Advanced Encryption Standard (AES). 2001.

[36] Chaim Perelman and L. Olbrechts-Tyteca. *The New Rhetoric: A Treatise on Argumentation*. University of Notre Dame Press,, 1969. trans. John Wilkinson and Purcell Weaver.

[37] Chris Reed and Glenn Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal of AI Tools*, 14:961–980, 2004.

[38] Chris Reed and Glenn Rowe. Araucaria 3.1 User manual. `http://araucaria.computing.dundee.ac.uk/downloads/version3_1/usermanual3_1.pdf`, 2006.

[39] Chris Reed, Douglas Walton, and Fabrizio Macagno. Argument diagramming in logic, law and artificial intelligence. *The Knowledge Engineering Review*, 22(01):87–109, 2007.

[40] B. Rolf and B. J\önsson. Athena : Argumentation from traditional humanities to IT. *Final report of Project 103/98*.

[41] Carolyn Rosé, Yi-Chia Wang, Yue Cui, Jaime Arguello, Karsten Stegmann, Armin Weinberger, and Frank Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning*, 3(3):237–271, 2008.

[42] Glenn Rowe, Fabrizio Macagno, Chris Reed, and Douglas Walton. Araucaria as a tool for diagramming arguments in teaching and studying philosophy. *Teaching Philosophy*, 29:111–124, 2006.

[43] Robert J Shimonski. Defining a security policy. `http://www.windowsecurity.com/articles/Defining\_a\_Security\_Policy.html`, 2003.

[44] S. B Shum et al. Sensemaking on the pragmatic web: A hypermedia discourse perspective. In *Proceedings of the 1st International Conference on the Pragmatic Web*, 2006.

[45] Borja Sotomayor. 10.2. transport-level and message-level security. http://gdp.globus.org/gt4-tutorial/multiplehtml/ch10s02.html.

[46] Andrew Stranieri, John Zeleznikow, and John Yearwood. Argumentation structures that integrate dialectical and non-dialectical reasoning. *Knowl. Eng. Rev.*, 16(4):331–348, 2001.

[47] L.G. Thomas and D.G. Knezek. Technology literacy for the nation and for its citizens. `http://www.iste.org/Content/NavigationMenu/Research/Reports/Technology_Literacy_White_Paper_1995_/Technology_Literacy_for_the_Nation_and_for_Its_Citizens.htm`, October 2005. from the International Society for Technology in Education web site.

[48] Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, Cambridge, 1958.

[49] Charles R. Twardy. Argument maps improve critical thinking. *Teaching Philosophy*, 27:95116, 2004.

[50] Jan M. van Bruggen, Henny P. A. Boshuizen, and Paul A. Kirschner. A cognitive framework for cooperative problem solving with argument visualization. pages 25–47, 2003.

[51] Frans. H. van Eemeren. *Crucial concepts in argumentation theory*. Amsterdam University Press, Amsterdam, 2001.

[52] Frans H. van Eemeren and Peter Houtlosser. Dialectic and rhetoric : the warp and woof of argumentation analysis. Kluwer Academic Publ., 2002.

[53] Tim van Gelder. What is argument mapping? `http://timvangelder.com/2009/02/17/what-is-argument-mapping/`, February 2009.

[54] Bart Verheij. Argue! - An implemented system for computer-mediated defeasible argumentation. In *THE TENTH NETHERLANDS/BELGIUM CONFERENCE ON ARTIFICIAL INTELLIGENCE (NAIC '98*, pages 57–66, 1998.

[55] Bart Verheij. ArguMed - A template-based argument mediation system for lawyers., 1998.

[56] Bart Verheij. Automated argument assistance for lawyers. In *Proceedings of the 7th international conference on Artificial intelligence and law*, pages 43–52, 1999.

[57] Bart Verheij. Artificial argument assistants for defeasible argumentation. *Artificial Intelligence*, 150:291–324, 2003. AI and Law.

[58] Bart Verheij. The Toulmin argument model in artificial intelligence. pages 219–238, 2009.

[59] R. Wash. Lecture notes on stream ciphers and RC4. *Reserve University*, pages 1–19.

[60] D. A Wheeler. Chapter 4. Security requirements. *URL http://www. dwheeler. com/secure-programs*, pages 31–36, 2003.

[61] O. Willmann. The seven liberal arts. In *The Catholic Encyclopedia*, New York, 1907. Robert Appleton Company. Retrieved April 14, 2010 from New Advent: `http://www.newadvent.org/cathen/01760a.htm`.

[62] Pan Yu and Han Jun. Linux-based network security administration policy system. http://www.ll2t.cn/16462.html.

Appendices

Appendix A

Details of User Interface Results

Table A.1: Araucaria 3.1 - Results about *User Interface Convenience*.

***User Interface Convenience* of Araucaria 3.1.**
How accurate is the information provided to the user (insufficient, good, too complicated) ?
9/10 ▷ The visual design of the user interface provides easy to read and easy to understand information.
Is the workspace correctly shared between panels ?
7/10 ▷ Yes, the most important panel is the graphical view and it gets the biggest place in the workspace. Limits between panels size are fixed and can not be expanded or contracted. In some cases, it would be interesting to enlarge the diagram panel but it is not possible.
Is the user operating interface intuitive and simple enough for users ?
5/10 ▷ The first transformation text selection in free premise is not intuitive. This asks to consult the help file. The basic principle is to select the premise text by holding down the mouse button and dragging from the beginning to the end of the premise in the textual panel, and for adding the selected premise to the diagram, just simply click once in the premise free panel in the right-hand diagram area. It is very simple but not intuitive because user first intuition is to select the corresponding text, drag it by mouse and drop it in the diagram panel.
Is the argument map assisted by software to get a better visual rendering ?
7/10 ▷ Yes, the map is displayed in an automatic way. User has no action available on the position of argument components in the map.
Is the interface appealing to users ?
5/10 ▷ The personalization of the panels background colour might contribute to attract the user but the lack of mouse actions in diagram building seems to be a major disadvantage.
Is the software tool designed to conceive large scale arguments ?
8/10 ▷ Yes, the *full size* zoom is conceived to display large size arguments in the window. Furthermore, the graphical panel turns into a scroll window in the presence of many arguments to allow navigation toward each components.
Does the interface allow the modification of argument maps ?
7/10 ▷ Modification of text and id is available functions, but the software does not allow to move argument boxes.
Does the software support zoom functions?
7/10 ▷ Yes, but not the common zoom with a percentage scale. The zoom function implemented in Araucaria is a specific zoom defined to adjust the whole argument in the window.
Does the software support drag-and-drop operations or point-and-click function on arguments elements with the mouse ?
3/10 ▷ The drag-and-drop is not implemented and the point-and-click function allows a restricted use.
Average score : 6.5/10

Table A.2: Athena Standard - Results about *User Interface Convenience*.

***User Interface Convenience* of Athena Standard.**
How accurate is the information provided to the user (insufficient, good, too complicated) ?
3/10 ▷ Information enabling the user to understand the functioning of the tool and its various features is considered insufficient. Utilization of this tool is mostly by trial and error for user without any recourse to a nonexistent manual. The lack of an information bar is really a disappointing aspect of the tool because any other mean is not used to replace it.
Is the workspace correctly shared between panels ?
7/10 ▷ Yes, graphical panel can be divided in many different open projects and provides a function for positioning all these different windows optimally.
Is the user operating interface intuitive and simple enough for users ?
9/10 ▷ Yes, software interface is really simple and intuitive enough for novice users.
Is the argument map assisted by software to get a better visual rendering ?
5/10 ▷ No, argument elements are only arranged on the panel by the user.
Is the interface appealing to users ?
5/10 ▷ Graphic windows have white background panels not customizable. Menu and toolbar style is near of the display of old Windows applications.
Is the software tool designed to conceive large scale arguments ?
7/10 ▷ Graphical window displays a scroll bar to allow the user to scroll up and down to view all of the argument. However, the tool does not provide any feature to mangage this complexity.
Does the interface allow the modification of argument maps ?
9/10 ▷ Yes, repositioning the argument elements with the mouse is intuitive and simple.
Does the software support zoom functions?
7/10 ▷ Yes, the zoom magnification is available from 10% to 100%.
Does the software support drag-and-drop operations or point-and-click function on arguments elements with the mouse ?
8/10 ▷ Yes, dragging and dropping are mainly used for arranging the current argument diagram, and pointing and clicking to alter elements properties. Nevertheless dragging and dropping do not permit to move a part of a project directly in the window of another project, all movements must remain within the window.
Average score : 6.5/10

Table A.3: ArguMed 3 - Results about *User Interface Convenience*.

***User Interface Convenience* of ArguMed 3.**
How accurate is the information provided to the user (insufficient, good, too complicated) ?
4/10 ▷ Information provided to the user is minimal. Some icons or menu items would require much more description. The lack of information is a major constraint augmented by the non-existence of a manual document.
Is the workspace correctly shared between panels ?
8/10 ▷ Yes, the graphical panel has the biggest space of the workspace and it is legitimate.
Is the user operating interface intuitive and simple enough for users ?
6/10 ▷ The user interface is poor which makes it simple. Intuitive operations such as double-clicking to edit statement content and the possibility to use the right-click of the mouse to appear specific menu
Is the argument map assisted by software to get a better visual rendering ?
8/10 ▷ Yes, the visual rendering is very clear thanks to the automatic disposition of statements.
Is the interface appealing to users ?
5/10 ▷ The user interface is common for this kind of tools. It is not design to be appealing but only useful.
Is the software tool designed to conceive large scale arguments ?
8/10 ▷ Yes, complex or large scale arguments are displaying in a very clear way with this tool.
Does the interface allow the modification of argument maps ?
7/10 ▷ Editing statements content is possible in the tool but moving boxes and links into the argument map is not an available functionality implemented. That said, the automatic positioning of arguments is very clear therefore the need to change map disposition is low.
Does the software support zoom functions?
0/10 ▷ No
Does the software support drag-and-drop operations or point-and-click function on arguments elements with the mouse ?
5/10 ▷ Only point-and-click function available.
Average score : 6 /10

Table A.4: Argumentative - Results about *User Interface Convenience*.

***User Interface Convenience* of Argumentative.**
How accurate is the information provided to the user (insufficient, good, too complicated) ?
8/10 ▷ Information provided by the tool is simple and in good quantity. Furthermore, if other issues remain, a quick detour in the help file allows to fill the gap.
Is the workspace correctly shared between panels ?
10/10 ▷ Yes, the graphical view in the workspace takes up the major part of the window. Furthermore, limits between panels can be adapted based on the specific use so panels size is very adjustable.
Is the user operating interface intuitive and simple enough for users ?
8/10 ▷ Yes, the user interface is simple and intuitive, very few difficulties encountered during the assessment.
Is the argument map assisted by software to get a better visual rendering ?
9/10 ▷ Yes many option parameters help to display correctly the argument map.
Is the interface appealing to users ?
10/10 ▷ Yes, Argumentative interface is defined in a modern style, with fresh icons and trendy diagram boxes.
Is the software tool designed to conceive large scale arguments ?
9/10 ▷ Yes, for example a zoom function is predefined to fit the whole diagram in the window. Furthermore, the argument tree view on the left of graphical view allows to display many argument components.
Does the interface allow the modification of argument maps ?
8/10 ▷ Yes, the edition panel is precisely created to enable the user to alter statements content. Besides this, a simple drag-and-drop operation allows to shift subargument to another statement.
Does the software support zoom functions?
10/10 ▷ Yes, many zoom functions are available.
Does the software support drag-and-drop operations or point-and-click function on arguments elements with the mouse ?
9/10 ▷ Yes all these functionalities are available in the user interface.
Average score : 9/10

Table A.5: Rationale 2 - Results about *User Interface Convenience*.

User Interface Convenience **of Rationale 2.**
How accurate is the information provided to the user (insufficient, good, too complicated) ?
10/10 ▷ Application's users can collect lot's of very complete information when using the software.
Is the workspace correctly shared between panels ?
10/10 ▷ Yes, the taskbar has a suitable size on the top of the interface, the building panel can be reduced or closed to let more space to the graphical view. The balance between the size of the panels can be adjusted according to user needs.
Is the user operating interface intuitive and simple enough for users ?
9/10 ▷ Yes, the interface is designed to be as intuitive and easy to use as possible.
Is the argument map assisted by software to get a better visual rendering ?
10/10 ▷ Yes, a special feature is implemented to reorganise complex maps in order to aid comprehension. The visual rendering is obviously better.
Is the interface appealing to users ?
9/10 ▷ Of course, it is. The interface design has been carefully crafted and is a joy to use for users. It has nice colours and is uncluttered and functional.
Is the software tool designed to conceive large scale arguments ?
9/10 ▷ Yes, a workspace overview is available to treat with large scale arguments.
Does the interface allow the modification of argument maps ?
8/10 ▷ Yes, drag and drop features are completly implemented and help users to change the relationships between argument elements on the workspace. The positioning feature does not permit the user to arrange the diagram's components where he wants, but it reorganises all boxes such as a pyramid structure for maximum visual clarity.
Does the software support zoom functions?
9/10 ▷ Yes, user can change the current zoom factor from 20 percent to 500 percent for precise viewing.
Does the software support drag-and-drop operations or point-and-click function on arguments elements with the mouse ?
10/10 ▷ Yes, mouse pointing and clicking is an effective way to navigate through the user interface, to selectively arrange the argument's structure and to handle some items
Average score : 9.33/10